

Прирачник за наставата по изборниот предмет

ПРОГРАМИРАЊЕ

*Маја Лошаноска - Тодоровска
Ирина Иванова*



Прирачник за наставата по изборниот предмет

ПРОГРАМИРАЊЕ

*Маја Лошаноска - Тодровска
Ирина Иванова*



БИРО ЗА РАЗВОЈ НА ОБРАЗОВАНИЕТО

Издавач:

Биро за развој на образованието

За издавачот:

Весна Хорватовиќ, *директор*

Редакција:

Иванка Мијиќ, *одговорен уредник*

Ајше Ајрулаи, *член*

Мирјанче Милошевска, *член*

Рецензенти:

Илија Лешковски

Митко Чешларов

Лектура:

Сузана Стојковска

Дизајн и корица:

Светлана Стојчева

Печати:

Киро Дандаро - Битола

Тираж:

400

Скопје, 2010 година

CIP - Каталогизација во публикација

Национална и универзитетска библиотека Св. „Климент Охридски“, Скопје

373.3.091.3:004.424(035)

ЛОШАНОСКА-Тодоровска, Маја

Прирачник за наставата по изборниот предмет Програмирање / Маја Лошаноска-Тодоровска,
Ирина Иванова, - Скопје : Биро за развој на образованието, 2010, - 49 стр. : 24 см

ISBN 978-600-206-018-7

1. Иванова, Ирина (автор)

а) Компјутерско програмирање - Основно образование - Наставни методи - Прирачници

COBIS.MK-ID 84438794

Програмирање е избран предмет кој може да се изучува во VII или VIII одделение на осумгодишното основно училиште, односно VIII или IX одделение на деветгодишното основно училиште со 72 часа годишно.

Предметот има за цел да ги запознае учениците од основното образование со основните концепти на програмирањето кое е суштински дел на информатиката како наука за работа со податоци и со тоа да им овозможи да ја согледаат разликата помеѓу корисниците на готови софтверски производи и креаторите на компјутерски програми. Ова сознание треба да им помогне на учениците во изборот на нивното понатамошно образование, а можеби и идната професија во животот.

Преку изучувањето на содржините по наставниот предмет учениците ќе го развијат логичкото размислување и ќе се стекнат со вештини за анализа на проблемски ситуации од секојдневниот живот и поставување најдобри и сеопфатни решенија за истите.

Во прирачникот се објаснети основните поими и концепти на програмирањето преку примери со два програмски пакета кои содржат некои од најосновните концепти од објектно-ориентираните програмски јазици: Scratch (визуелен програмски јазик) и SmallBasic (програмски јазик со класично пишување на програмски код).

Во зависност од нивото на знаење и интереси на учениците кои ќе изберат да го изучуваат предметот, наставникот може да ја реализира наставната програма со тоа што ќе работи одреден број на часови во двата програмски јазици или сите часови може да ги реализира со работа во еден од програмските јазици. Исто така, наставникот може да одлучи да работи и со друг програмски јазик во зависност од интересите на учениците и расположливиот хардвер и софтвер.

За секоја тема од годишната програма за работа во прирачникот има разработени примери на настав-

ни содржини со цели, методи и активности, објаснување на новите поими и концепти во врска со програмскиот јазик. Наставникот по свој избор може да ги рedefинира наставните содржини според условите во кои ја реализира наставата по предметот програмирање. Исто така, наставникот при реализација на наставните содржини треба да им понуди на учениците свои примери на интересни, креативни и поттикнувачки проблемски ситуации за кои треба да се изработат соодветни програмски решенија.

Наставни средства кои може да се користат во реализација на наставата се: персонален компјутер, проектор, локална компјутерска мрежа, Интернет, наставни ливчиња, образовни софтвери, списанија, аудио-визуелни средства и сл.

НАЧИН НА ОРГАНИЗАЦИЈА НА РАБОТАТА НА ЧАСОТ

Начинот на работа зависи од условите и расположливите средства во кабинетот по информатика. Се препорачува организирање на наставата во блок-часови. Учениците треба да работат самостојно или најмногу во група од двајца ученици. Секоја изработка на часот ученикот треба да ја меморира во својот училиштен фолдер.

НАЧИН НА ОЦЕНУВАЊЕ

Формативно: Секој ученик има свое портфолио кое содржи показатели (ученички изработки на компјутер) и однапред подготвени евалвациони листи кои се изработуваат по конкретна негова активност.

Сумативно – Се изработуваат тестови на знаења и/или проектни задачи на крајот од секој квалификационен период.

Од авторите

Наставни содржини од ТЕМА 1: ПОИМ ЗА АЛГОРИТМИ И ПРОГРАМИ

Алгоритми и нивно претставување

Цел на часот:

- ученикот/ученичката да го објаснува поимот алгоритам;
- да ги осознае основните концепти на алгоритамското размислување;
- да се оспособува за изработка на едноставни алгоритми

1 час

Воведен дел

Дискусија: Што е заедничко на: вртењето телефонски број, рецептот за правење торта, варењето чај, купувањето леб? Тие се или вклучуваат листа од инструкции.

Всушност сè што правиме може да се запише постапно во вид на чекори/инструкции.

Главен дел: Наставникот презентира пример од секојдневниот живот запишан постапно, во чекори.

Активност 1



Телефонирање

Чекор 1: Дигни ја слушалката.

Чекор 2: Сврти го бројот.

Чекор 3: Почекај да се јави личноста.

Чекор 4: Зборувај со личноста.

Чекор 5: Спушти ја слушалката.

Активност 2

Рецепт за чоколадна торта

Состојки: 400 г. чоколада, 3 јајца, 1 маргарин, 1 пакетче ванилин шеќер, 2 чаши шеќер, 1 чаша брашно.

Чекор 1: Стопи ги чоколадата и маргаринот.

Чекор 2: Истури го шеќерот во стопената чоколада и измешај.

Чекор 3: Истури ги јајцата и ванилата и измешај.

Чекор 4: Истури го брашното и измешај.

Чекор 5: Добиената смеса истури ја во тавче.

Чекор 6: Печи на 250 степени околу 40 минути.

Чекор 7: Се остава да се излади и се јаде.

Наставникот го запознава учениците со поимот алгоритам.

Алгоритам претставува множество упатства за извршување операции со точно зададен редослед со цел решавање на даден проблем.

Активност 3

Самостојна работа - наставникот им задава на учениците алгоритам (да се нацрта слика) кој треба да го следат, без претходно да го објасни крајниот резултат. Секој ученик работи индивидуално и при тоа не треба да гледа во изработките на другите ученици, ниту, пак, да побара дополнително објаснување.

Алгоритам за цртање чаша за чај

1. Повлечи права линија.

2. На средината нацртај мал круг.

3. Нацртај друга права линија паралелна со првата.

4. Нацртај права линија која ќе ги поврзува долните врвови на двете претходно нацртани прави линии.

5. Нацртај елипса која ќе ги допира горните врвови на двете паралелни линии.

Учениците покажуваат што нацртале. Дискусија: Дали сите имаат ист алгоритам? Дали сите нацртаа иста слика? Зошто? Што беше тешко при следење на упатствата? Што им недостатуваше на упатствата? Кои карактеристики треба да ги има добриот алгоритам?



Алгоритмите треба да се недвомислени, јасни, прецизни - секоја операција треба да е јасно определена, треба прецизно да е утврден редоследот на извршувањето на операциите и треба да е разбирлив за секого, независно кој го напишал. За даден проблем може да се напишат различни алгоритми и сите

тие да бидат правилни, т.е. да даваат точно решение.

Завршен дел:

Учениците изработуваат едноставни алгоритми (во корелација со друг наставен предмет).

2 час

Воведен дел

Дискусија: Што е алгоритам? Зошто воопшто се учат алгоритмите? Кои карактеристики треба да ги има алгоритмот? Кои се чекорите за решавање на еден проблем?

Чекори за решавање проблем:

1. сфати го проблемот;
2. размисли како можеш проблемот да го претставиш со алгоритам;
3. формулирај го алгоритмот;
4. провери ја точноста на алгоритмот.

Главен дел

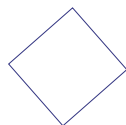
Алгоритмот може да се прикаже на два начини: текстуално (природен јазик, псевдојазик-јазик сличен на природниот, со попрецизна синтакса и семантика) и графички.



Блок за почеток и крај на алгоритмот



Блок за аритметички операции



Блок за логичко одлучување



Конектор (влез или излез во друг дел на дијаграмот, пример: поврзување на делови на дијаграм кој се наоѓа на различни страници).



Блок за влез на податоци



Блок за излез на податоци



Блок за поврзување

Графички симболи кои најчесто се користат за приказ на алгоритмот се:

Наставникот презентира едноставен алгоритам кој е претставен графички.

Дискусија: Кои се предностите и недостатоците при графичкото претставување алгоритам?

Графичкото претставување дава поголемата прегледност на текот на дејствата во алгоритмот. Графичкото претставување е непогодно за поголем алгоритам кој може да зафати повеќе страници, во кои тешко би се снаоѓале.

Завршен дел

Учениците изработуваат едноставни алгоритми со графички приказ (во корелација со друг наставен предмет).

ПОИМ ЗА ПРОГРАМА КАКО ДЕЛ ОД СОФТВЕРОТ

Цел на часот:

- ученикот/ученицката да го објаснува поимот програма;
- да ја сфати разликата помеѓу алгоритам и програма.

Воведен дел: Повторување: Да се потсетиме - Што е податок? Што е софтвер? Што е алгоритам?

Главен дел: Наставникот го воведува новиот поим – програма.

Компјутерите не работат ништо без некој да им каже што да прават. За компјутерот да направи нешто, мора да му се дадат инструкции – чекор по чекор.

Програма претставува множество (низа, листа) од инструкции за решавање на одреден проблем со компјутер.

Системските програми управуваат со работата на компјутерот. Апликативните програми се наменети за извршување на конкретна обработка на податоците со компјутер кои уште се нарекуваат апликации.

Дискусија: Дали програмите се дел од софтверот? Кои се разликите помеѓу алгоритам и програма? Кои програми ги познаваш?

Алгоритмите се поопшти и поапстрактни од програмите. Еден алгоритам може да биде решен од човек или машина, или од двете. Програмата мора да биде извршена од компјутер. Врската помеѓу програма, алгоритам и податоци е дадена во формулата на Niklaus Wirth (автор на јазикот Pascal)

Програма = Алгоритам + Податоци

Завршен дел

Наставникот задава одреден проблем. Учениците треба да направат алгоритам за негово решавање. Потоа треба да ги идентификува дополнителните прашања потребни при создавање соодветна програма.

Пример:

Проблем: Задавање упатство каде се наоѓа кабинетот по информатика во училиштето.

Чекор 1: Влези во училиштето на главниот влез.

Чекор 2: Оди право.

Чекор 3:

При создавање програма мора да се даде одговор на прашањата:

1. Како ќе се идентификува главниот влез во училиштето?
2. Колку треба да се оди право?
3.

УЛОГАТА НА ПРОГРАМИТЕ ВО КОМПЈУТЕРОТ

Цел на часот:

ученикот/ученичката да ја осознае примената на програмите во компјутерите и секојдневниот живот.

Воведен дел

Дискусија: Што е задача на програмите? Кои програми учениците ги користеле при работата со компјутер?

Програмите му кажуваат на компјутерот како да реши одреден проблем. Всушност, задача на програмите е да му кажат на компјутерот како да ги прифати влезните податоци, да оперира со нив и да врати излезни податоци.

Главен дел

Активност 1

Наставен лист (на учениците им се дава празна табела која самостојно треба да ја пополнат):

Тип на програма	Влез	Што работи програмата	Излез
за обработка на текст	карактери внесени од тастатура	форматирање на текст	Прикажан или испечатен уредно организиран текст
Игра	од тастатура	пресметува колку брзо и колку далеку да ја движи фигурата на екран	фигурата се движи на екран
за оптичко препознавање на знаци	текст од скенер	ги препознава формите на карактерите	Конвертира скениран текст во текст документ кој може да се обработува со програма за обработка на текст
...			

Учениците ги презентираат пополнетите табели при што се развива дискусија.

Активност 2

Дискусија: Дали само компјутерите користат програми? Каде во секојдневниот живот се среќаваме со програми?

Програмите не се користат само во работата со компјутер. ТВ - уредите, музичките системи, клима уредите можат да се програмираат и со нив да се управува од одредено растојание. Тоа значи дека овие уреди користат програми кои се вградени. Пример: телевизорот за да може да ги разбере влезните команди што ги испраќа корисникот преку далечинскиот управувач мора да има вградено програма. Врз основа на добиената команда, чипот во кој е испрограмирана логиката на телевизорот иницира одредена активност, т.е. поттикнува други електронски компоненти.

Завршен дел

Секој ученик треба да даде пример како програмите се составен дел од нашиот живот.

Наставни содржини од ТЕМА 2: ПРОГРАМИРАЊЕ, ПРОГРАМСКИ ЈАЗИЦИ И ИНТЕГРИРАНИ ОКОЛИНИ ЗА ПРОГРАМИРАЊЕ

Програмирање и програмски јазици

Цел на часот:

- ученикот/ученичката да се запознае со концептот на програмирање;
- да го објаснува поимот програмски јазици, програмирање, програмер;
- да се запознае со различните типови на програмски јазици.

Воведен дел

Дискусија: Како комуницираат луѓето? Што е тоа природен јазик? Кои природни јазици учениците ги знаат? Дали слушнале за вештачки јазик и дали знаат пример? Дали природните јазици се добри за комуникација човек – компјутер? Зошто?

Јазикот е средство кое овозможува комуникација помеѓу луѓето. Природните јазици (македонски, англиски, руски, француски) настанале спонтано во дамнешни времиња и постојат група луѓе (народи, нации, племиња и сл.) кои го разбираат, зборуваат и размислуваат на него, односно го користат. Но, природните јазици поради својата нееднозначност и непрецизност се недоволно строги и не се погодни за претставување на специфични, на пр. математички информации. Затоа се измислени вештачки јазици. Пример за вештачки јазик: јазици за бележење на математички, физички или хемиски формули, јазикот есперанто итн.

Главен дел

Наставникот ги објаснува новите поими – програмски јазици, програмирање, програмер.

За да работат компјутерите потребни се програми. Програмите се напишани на јазици кои се специјално дизајнирани, со ограничено множество на инструкции. Овие јазици се наречени програмски јазици.

Програмски јазик е вештачки јазик и се користи за пишување програми, односно софтверот на компјутерот.

Компјутерско програмирање се дефинира како процес на создавање на компјутерска програма со користење на програмски јазик.

Во почетокот програмските јазици личеле и по логиката на програмирањето биле слични со машинскиот јазик (машинскиот јазик се состои од наредби кои се изразуваат само со нули и единици т.е. само со бинарни цифри). Тоа се симболички програмски јазици. Наместо да се учи мајчиниот јазик на компјутерот (машински јазик), развиени се виши програмски јазици кои имаат приказ на наредби слични на секојдневниот јазик. Пример: Fortran, Basic, Pascal, C (C++), Visual Basic, Java.

Наставникот презентира примери на програмски јазици.

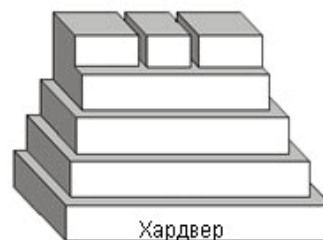
Луѓето кои ги креираат (пишуваат) компјутерските програми се викаат програмери.

Завршен дел: Повторување на новите поими.

Пример за вежба од наставен лист

Вметни ги поимите соодветно во цртежот

виши програмски јазик Pascal C
симболички јазик Java машински јазик



ПРОЦЕС НА ИЗРАБОТКА НА ЕДНА ПРОГРАМА (ОД ИЗВОРНА ДО ИЗВРШНА ВЕРЗИЈА)

Цел на часот

- ученикот/ученичката да го осознае процесот на пишување и извршување на една програма;
- да ги познава и правилно да ги именува и користи елементите од интегрираната околина за програмирање (едитор, компајлер, дебагер (покажувач на грешки и помош)).

Воведен дел

Голем број проблемски ситуации и задачи денес се решаваат брзо и ефикасно со помош на компјутер. Чекорите за нивното решавање мора да бидат запишани во вид на програма во некој програмски јазик. Изработката на компјутерска програма која решава одредена задача или проблемска ситуација е процес кој се одвива низ неколку фази.

Главен дел

Фази во процесот на изработка на компјутерска програма:

1. Анализа на проблемската ситуација и разгледување можни решенија.

2. Алгоритамско претставување на чекорите за можното решение на проблемската ситуација (графички или со псевдо код).

3. Кодирање на алгоритамот.

Запишување на алгоритамот како низа од инструкции во избран програмски јазик. Вака запишаниот алгоритам претставува програмски код кој се вика *изворна програма*.

4. Преведување на изворната програма во извршна програма

Изворната програма не е разбирлива за компјутерот. Таа мора да се преведе во машински јазик (во кој инструкциите се запишуваат со 1 и 0) за да биде разбрана и извршена од процесорот. Изворната програма преведена на машински јазик се вика *извршна програма*.

За преведување на изворната програма во извршна програма се користат *програми преведувачи* кои во зависност од начинот на преведување може да бидат *компајлери (најчесто)* и *интерпретери*.

5. Корекција на грешки (синтаксни, логички)

Преведувањето на изворната во извршна програма ретко кога е без грешки. Програмата преведувач го алармира програмерот за направените грешки во изворната програма кои мора да бидат коригирани за инструкциите да можат да се преведат во машински јазик. *Синтаксните грешки* се однесуваат на неправилно напишани зборови од програмскиот јазик, неправилно користени или испуштени интерпункциски знаци и сл. и најчесто се поправаат при преведувањето на изворната програма со помош на преведувачот.

Логичките грешки треба да ги открие и поправи програмерот при тестирањето на извршната програма. Логичките грешки обично се викаат багови (bugs) и за нивно откривање се користи програма дебагер која овозможува извршување на програмата линија по линија и стопирање на нејзиното извршување на одредено „сомнително“ место.

По корекција на грешките програмата повторно се преведува. Овие чекори се повторуваат сè додека програмата има грешки и завршува кога програмерот е задоволен од резултатите на тестирањето.

Пример за синтаксна и логичка грешка

Ова е равенка за периметар на правоаголник ако се познати неговите страни:

$$L=2*(a-b)$$

Синтаксна грешка – недостасува заграда на крајот од изразот.

Логичка грешка – формулата не е точна иако по ставање на заградата на крајот е правилно запишана.

$$a=5$$

$$b=\text{“MaJa”}$$

$$c= b-a$$

Логичка грешка – b е текст и не може да биде намаленик.

6. Тестирање на програмата

Тестирањето се прави од страна на програмерот или корисниците за кои е наменета програмата. Програмата мора да работи правилно за различни вредности кои го внесува корисникот или да оневозможи погрешно внесување вредности во програмата преку пораки и сл. до корисникот.

Пример за тестирање

Како би можело да изгледа тестирање на едноставна програма која дели два внесени броја од корисникот и го прикажува резултатот:

Тест 1: Корисникот внесува нула како делител

Тест 2: Корисникот внесува текст наместо број и сл.

Тест 3: Корисникот внесува само еден број

Бидејќи делење со нула не е дозволено и делењето е операција која работи со броеви програмата треба да има соодветно решение за овие случаи преку испишување порака до корисникот, враќање на повторни внесување и сл.

По успешното поминување на сите фази од процесот може да се каже дека програмата е готова и работи правилно.

Завршен дел

Активности за практична примена на усвоените поими од часот.

Активност:

Претстави го целиот процес на програмирање (неговите фази) во вид на алгоритам (графички или со псевдокод).

Спореди: скица за нова куќа, проект на куќата од архитект според скицата, изградба на куќата од градежна фирма според проектот на архитектот со: алгоритам, изворна и извршна програма.

Пронајди каде има синтаксички, а каде логички грешки во следните реченици:

Ќе одам во супермаркет да купам леб млеко кафе

Вчера ќе одам во парк да возам велосипед.

Утр ќе одам парк да возам веосипед

Како би тестирал програма која пресметува катета на правоаголен триаголник ако корисникот внесува вредности прво за втората катета, а потоа за хипотенузата?

(Хипотенузата мора да биде поголема од катетата, вредностите треба да бидат само позитивни и различни од нула и сл.)

ЗАПОЗНАВАЊЕ СО ОСНОВНИТЕ ЕЛЕМЕНТИ НА ИНТЕГРИРАНАТА ОКОЛИНА ЗА ПРОГРАМИРАЊЕ

Внесување и извршување на готов пример програмски код

Цел на часот:

- ученикот/ученичката да се оспособи да извршува готов точен програмски код;
- да ја осознае функцијата на дебагерот при извршување на неточен програмски код;
- да ги препознава и да ги разликува датотеките кои се креираат при извршување и зачувување на програмата.

Воведен дел

За да ги пишува и развива своите програми програмерот користи група на програми кои ја сочинуваат таканаречената *околина за програмирање*. Програмската околина речиси секогаш се состои од едитор, компајлер, библиотека на готови програми и дебагер. Овие програми може да бидат посебни или меѓусебно поврзани – интегрирани во една програма која се вика *интегрирана развојна околина* (Integrated Development Environment - IDE) за програмскиот јазик.

Програмски едитор е специјално дизајниран текстуален едитор во IDE каде се пишува изворниот код на програмата. Најчесто програмските едитори овозможуваат синтаксичко маркирање во различна боја, автоматско порамнување на исказите, автоматско комплетирање при пишување на исказите, синтаксни проверки, спарување на загради и сл.

Библиотека на готови програми претставува фајл кој е колекција од веќе напишани мали често користени програми во програмската околина. Има графички, математички библиотеки и сл. Овие библиотеки програми може да бидат повикани и извршени од новонапишаната програма со што им се олеснува на програмерите и се скратува времето на развивањето на новата програма.

Главен дел

Разгледување и објаснување на значењето и функцијата на елементите на развојната околина.

Активност 1: Стартување на програмата за програмирање и разгледување на нејзините основни елементи. Практично извршување на готови пример програми.

Програмски јазик Скреч

Скреч (Scratch) е визуелен виши програмски јазик со кој се креираат интересни приказни, анимации, игри. Скреч е и името за графичкото опкружување за развој и публикување на програмите пишувани со програмски јазик Скреч. Програмирањето во Скреч се базира врз правење на проекти со помош на објекти, наречени фигури.

Наставникот ги објаснува правилата за градење програма – блокови кои можат да се спојат заедно во множества и извршување на програма.

Активност 2: Програмски јазик Small basic

Пишување на готова програма со синтаксни и логички грешки (редовите со зелена боја). При пишување на програмата треба да се воочи и објасни синтаксното маркирање, автоматското порамнување и можноста за автоматското комплетирање. Воочување на контекстно осетливиот помошник (Help).

```
TextWindow.BackgroundColor = "red"  
TextWindow.ForegroundColor, "yellow"  
(недостасува =)  
TextWindow.Write („Kako ne se vikas?“)  
(две загради и логичка грешка)  
name = TextWindow.Read()  
TextWindow..Write(„Zdravo „ + name)  
(две точки наместо една)  
TextWindow.WriteLine(„ Jas sum small basic! Drago mi  
e!“)  
(Writelin наместо Writeline)
```

Дали е ова извршен или изворен код?

Извршување на програмата (Run - алатка, F5 тастер).

Која програма од околината за програмирање се активира при избор на наредбата за извршување на програмата?

Воочување на пораките за грешки. Во кои линии и кој ред од линијата во програмата има грешка? Дали е грешката синтаксна или логичка? Што треба да се направи по корекцијата на грешките?

Воочи: по корекција на грешките потребно е повторно извршување – компајлирање на програмскиот код. Каде се прикажуваат резултатите од програмата? Воочување на прозорец за излезните резултати од програмата.

Завршен дел

Активност 3: Меморирање на програмата со избор на ново име и локација.

Програмски јазик Small basic

Колку фајлови се креираат при меморирање на програмата? Која е нивната наставка? Ако се повеќе фајлови, дали и по што се разликуваат? Пробај да ги отвориш со двоен клик и одговори! Воочување на фајловите кои се креираат при меморирање (ако се повеќе идентификување на изворен фајл- *.sb и извршен фајл - *.exe, *.dll – фајл - библиотека на мали готови прорами – Dynamic Link Library).

Наставни содржини од ТЕМА 3: ПРОГРАМА СО РЕДОСЛЕДНА СТРУКТУРА

Основни елементи на програмскиот јазик. Искази за приказ на екран со техника на редоследно извршување

Цел на часот:

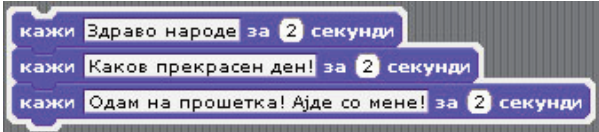
- ученикот/ученичката да ги осознае и објаснува основните елементи од програмскиот јазик (ПЈ);
- да ја осознае и да ја користи техниката за приказ на екран;
- да ја осознае техниката на редоследно извршување на секвенца од искази (напластени искази, напластување).

Воведен дел: Дискусија: Во меѓусебната конверзација луѓето користат зборови кои ги групираат во реченици. Со една или повеќе реченици во одреден редослед разменуваат информации и меѓусебно се разбираат.

Програмските јазици се состојат од точно одредена група на зборови кои програмерот според правилата ги подредува во искази (реченици) кои му пренесуваат информација на компјутерот.

Главен дел: Внесување на пример програма и нејзина анализа и дискусија за запознавање со градбените делови на програмскиот јазик и некои основни објекти, операции и својства.

Активност 1: Внесување на пример програма

Програмски јазик Скреч	Програмски јазик Small basic
 <p>Објасни: блокот „кажи...за ...секунди“ овозможува прикажување на говорен балон за дефинирано време.</p>	<pre>TextWindow.WriteLine („Zdravo narode!") TextWindow.WriteLine („Kakov prekrasen den!") TextWindow.WriteLine („Odam na prosetka! Ajde so mene!")</pre>

Од колку линии се состои програмскиот код? Дали разликуваш делови во линиите? Кои?

Обиди се да направиш споредба на градбените делови од програмскиот јазик со градбените делови на говорниот јазик.

Кои градбени делови од програмскиот јазик ги препознаваш во програмата?

Воочи: Секоја линија во програмата претставува исказ, а исказот претставува инструкција за работа на компјутерот. Секој исказ може да биде поделен на сегменти (делови) кои имаат одредено значење.

Градбени делови на програмскиот јазик се:

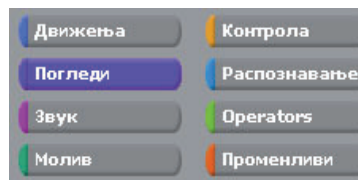
- зборови (клучни зборови од јазикот – резервирани зборови со веќе дефинирано значење, имиња на променливи, објекти, операции, својства и сл.);
- оператори (математички, логички, релациони, за доделување);
- интерпункциски знаци (точки, наводници, загради и сл.);
- коментари (произволен текст од програмерот за објаснување на текот на програмата).

Активност 2:

Изврши ја програмата. Анализирај ги прикажаните резултати од изворната програма.

Каде се прикажаа резултатите? Кој исказ се изврши прв? Промени го редоследот на исказите. Изврши ја програмата. Кој исказ сега се изврши прв? Дали има исказ кој не се изврши? Што можеш да заклучиш?

Програмски јазик Скреч



Во Скреч исказите се претставени во облик на блокови кои се поделени во 8 категории – движења, погледи, звук, молив, контрола, распознавање, оператори и променливи. Тие овозможуваат доделување инструкции на објектот - фигура. Наставникот ги објаснува категориите искази, како и начинот на нивното активирање и операциите со нив (бришење, дуплирање). За да се изврши исказ треба соодветниот блок да се извлече во областа со скрипти (програми) и потоа двојно да се кликне на него. Резултатот по извршувањето на програмата се прикажува во просторот – Етапа.

Програмски јазик Small basic

Воочи: TextWindow е текстуален прозорец во кој се прикажа резултатот по извршување на програмата и се вика објект во програмата. WriteLine е име на активност за пишување текст или број во текстуалниот прозорец, и се вика операција за објектот текстуален прозорец. Воочи дека испишувањето на следниот текст започнува во нова линија.

Текстот кој го испишува операцијата за пишување текст се вика влезна вредност за операцијата. Пишување на искази во програмскиот код кои се извршуваат сите и редоследно се вика техника на напластени искази.

Активност 3:

Програмски јазик Small basic

На почетокот од програмата додади ги следните искази:

```
TextWindow.BackgroundColor="red"
TextWindow.ForegroundColor="yellow"
```

Изврши ја програмата. Анализирај го приказот во прозорецот. Размисли што ќе се случи ако втората линија ја ставиш последна во програмата. Дали ќе се промени во жолта бојата на текстот? Зошто?

Промени ја вредноста за бојата во двете линии по желба. Промени го текстот што се испишува во прозорецот. Изврши ја програмата. Меморирај ја програмата во својот фолдер.

Кои разлики и сличности ги воочуваш во начинот на испишување на операциите и својствата во изворната програма? Воочи: *ForegroundColor* и *BackgroundColor* претставуваат имиња на својства на објектот *TextWindow*. Преку придружените вредности својства ја определуваат бојата на текстот и бојата на позадината во прозорецот. Својствата го определуваат изгледот на објектот преку доделените вредности. Во исказите по операцијата обично има загради со или без влезни податоци, а по својствата следува знакот еднакво и вредност за својството во наводници.

Завршен дел

Дискусија, анализа и споредба на предмет од секојдневниот животот со градбени делови од програмски јазик. Објект – велосипед; својства: боја, брзини; операции: вртење, притискање; Вредности за својствата: црвена, пет; влезни вредности за операциите (кочница, свонче)

Направи споредба со објект, својства, вредност на својствата, операции од ПЈ со велосипед, во црвена боја, со пет брзини, вртење на педалите, воланот, притискање на кочница, свонче.

ИЗРАБОТКА НА ПРОГРАМИ СО НАПЛАСТЕНИ ИСКАЗИ ЗА ПРИКАЗ НА ЕКРАН

Цел на часот:

- ученикот/ученичката да се оспособи за изработка на едноставна програма со редоследна структура, со напластени искази за приказ на екран.

Воведен дел: Повторување: Што се напластени искази? Во програмскиот јазик што претставува прозорецот во кој се прикажуваат излезните резултати? Која операцијата ја користевме во пример програмите за пишување во текстуалниот прозорец?







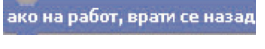
Главен дел:

Анализа на пример програми со техника на напластени искази.

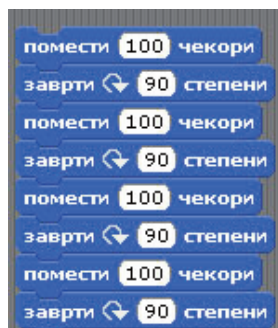
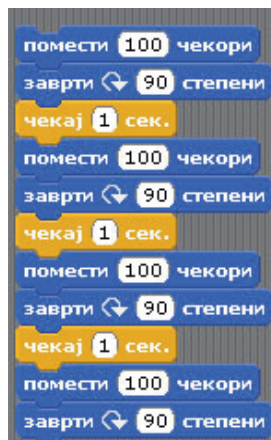
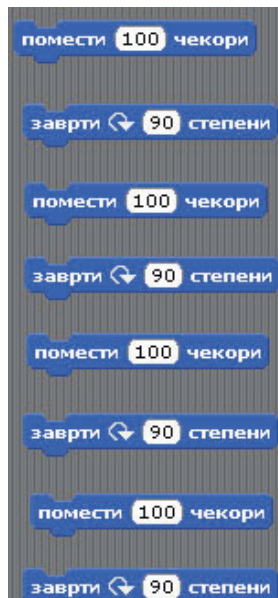
Активност 1:

Програмски јазик Скреч

Запознавање со основните блокови за движење на фигура.

Исказ (блок)	Значење
	Поместување на фигурата напред или назад.
	Ротирање на фигурата во насока на стрелките на часовникот.
	Ротирање на фигурата спротивно од насоката на стрелките на часовникот.
	Поставување на фигурата во дефинирана насока (0= нагоре, 90= надесно, -90= налево, 180= надолу)
	Поставување на фигурата во правец на стрелката или друга фигура.
	Поместување на фигурата на локација на стрелката или друга фигура.
	Свртување во обратна насока на фигурата кога таа ќе го допре работ на Етапата.

Наставникот презентира пример програми. Објаснува изработка на програма (пр. движење на фигура во правец на стрелките на часовникот). Потребните искази ги влече во областа со скрипти (програми) и потоа го извршува секој блок поединечно (двоен клик на секој блок). Креира скрипта (програма) со лепење на блоковите меѓусебно. Со двоен клик, каде било на множеството, започнува извршување на блоковите од почеток до крај. Се дискутира за добиениот резултат. Дали може да се подобри програмата? Го објаснува исказот “чекај ...секунда”.



Програмски јазик Small basic

Запознавање со нови операции и својства во текстуалниот прозорец. На пример, својство Title, операција Pause() и операција Clear(). Внеси го следниот програмски код со наплатени искази со помош на автоматското комплетирање:

```

TextWindow .Title ="PROGRAMCE"
TextWindow .BackgroundColor="green"
TextWindow .ForegroundColor ="black"
TextWindow.WriteLine („Zdravo!")
TextWindow.WriteLine („Jas sum tvojata prva programa!")
TextWindow.WriteLine („Moeto ime e PROGRAMCE.")

```

Изврши ја програмата. Дали текстуалниот прозорец во горната лента има име? Со кој исказ се доделува име на прозорец? Пронајди го делот што го доделува името. Дали според начинот на кој е запишан е објект, својство или операција? Користи го помошникот за да учиш за нови објекти и операции.

Дали ќе му се додели името на прозорецот ако исказот за доделување име од прва се премести во последната или некоја друга подолна линија од програмата? Размисли, провери!

Воочи: Title (име) е својство на текстуалниот прозорец кое го одредува неговото име.

Активност 2: Направи промени во претходниот пример програма.

Програмски јазик Small basic

Додај ги на крај следните искази:

```

TextWindow .BackgroundColor="white"
TextWindow .WriteLine („Za da se izbrise ekranot pritisni bilo koe kopce od tastaturata")
TextWindow .Pause ()
TextWindow .Clear ()

```

Што е ново и различно од претходните примери? Воочи ги новите операции во објектот TextWindow Pause() и Clear(). Дали овие операции имаат влезни вредности?

Дали програмата ќе го има истиот краен приказ и без исказот: TextWindow .WriteLine („Za da se izbrise ekranot pritisni bilo koe kopce od tastaturata")?

Размисли што ќе се случи ако во програмскиот код го избришеме исказот за пауза. Провери и одговори. Воочи: Со својството Title се доделува име на прозорецот, операцијата Pause() чека на акција (притискање тастер) од корисникот за да продолжи со текот на програмата. Операцијата Clear() го брише претходниот приказ во прозорецот.

Активност 3:

Програмски јазик Small basic

Промени во пример програмата за запознавање со операцијата Write со слична функција на Write-line.

Замени ја операцијата Writeline со Write. Изврши ја програмата, анализирај го приказот во прозорецот и спореди го со претходниот. Воочи дека Write е нова операција која како и Writeline пишува во текстуалниот прозорец но, овозможува следниот текст да биде испишан во иста линија со следниот.

Активност 4:

Програмски јазик Small basic

Внеси промена во програмскиот код за исказот со бела позадина да се испише во нова линија во прозорецот. Текстот за крај на програмата „Press any key to continue ...“ нека се испише во нова линија.

Завршен дел: Изработка на едноставни програми со редоследна структура со напластени искази.

Наставни содржини од ТЕМА 4: ПРОМЕНЛИВИ И ИСКАЗИ ЗА ДОДЕЛУВАЊЕ

Аритметички операции, изрази и променливи во програмскиот јазик

Цел на часот:

- ученикот/ученичката да го осознае концептот аритметички операции и изрази;
- да го објаснува поимот за променлива во програмски јазик;
- да ја осознае техниката за доделување на вредност;
- да се оспособи за користење на променливи и искази за доделување во програмскиот јазик;

• да се запознае со концептот за тип на променлива.

Воведен дел:

Дискусија за разликата помеѓу текстуален израз и броен (математички) израз и дефинирање на математички израз:

Која е разликата помеѓу следните изрази: Два+три, "2+3", 2+3 ?

Аритметички израз е бројна вредност или запис од две или повеќе бројни вредности и математички оператори меѓу нив. Математички оператори во програмскиот јазик се: +, -, *, /.

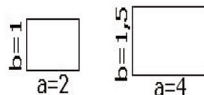
Главен дел:

Анализа на пример програми за воочување на разлики меѓу аритметички и текстуални изрази во програмски код, запознавање со поимот променлива во програмскиот јазик, правила за именување на променлива, доделување вредност на променлива и користење променливи во програмскиот код.

Променлива претставува име или симбол чија вредност може да се менува во текот на извршувањето на програмата. Така, на пример, во науката симболот T е ознака за температура. T е променлива.



Да земеме пример од математиката. a и b се страни на правоаголникот. Тие се променливи. За првиот правоаголник тие имаат една вредност, а за вториот тие се менуваат и имаат друга вредност. Но значењето им е исто - и во двата правоаголника a и b се страни на правоаголникот, а P е плоштина. Променлива во програмскиот јазик е место каде што привремено се сместуваат вредности кои подоцна може да се користат во програмата. Вредности кои и може да и се доделуваат на променливата се текст, број, вредност на математички израз или исказ која е исто број или текст. Симболот „=" е оператор за доделување вредност на променлива. Променливите имаат имиња. Имињата се доделуваат според правилата на програмскиот јазик.

$$P = a * b$$



Активност 1: Собирање на 2+3 и анализирање на исписот.

(Во програмскиот јазик Скреч се одбира дали променливата е за сите или само за една фигура.)

Програмски јазик Скреч	Програмски јазик Small basic
	<ol style="list-style-type: none"> 1. TextWindow.WriteLine („Sobiranje na dva broja:“) 2. TextWindow.Write („2+3=“) 3. TextWindow.WriteLine (2+3)
	<p>Провери што ќе се случи со излезот ако во линија 2 користиме WriteLine наместо write!Воочи: Во WriteLine текстот е испишан во наводници, а аритметичкиот израз без наводници.</p>

Активност2:

Програмски јазик Скреч

Пресметај плоштина на правоаголник со страни 3 и 5.



Програмски јазик Small basic

Спореди ги и анализирај следните програми.

<pre>TextWindow.WriteLine („2+3=“) TextWindow.WriteLine (2+3)</pre>	<pre>broj1=2 broj2=3 zbir= broj1+broj2 TextWindow.WriteLine („2+3=“)</pre>
<pre>TextWindow.WriteLine (zbir) TextWindow.WriteLine („Zdravo Maja“) TextWindow.WriteLine („Zdravo Gorjan“)</pre>	<pre>ime1=„Maja“ ime2=„Gorjan“ TextWindow.WriteLine („Zdravo "+ ime1) TextWindow.WriteLine („Zdravo "+ ime2)</pre>

Воочи: Операцијата WriteLine или Write може да имаат повеќе од една влезна вредност. Меѓу нив се поставува операторот „+“. Вредностите се пишуваат една до друга во иста линија во текстуалниот прозор. Воочи ги зборовите broj1, broj2, zbir, ime1, ime2 во програмскиот код. Ова се таканаречени променливи или варијабли.

Завршен дел: Самостојна изработка на програма со зададен излез. Давање насоки за работа:

Користи броеви по твој избор и смести ги во променливи. Користи променливи и за сместување на математичките изрази. Додели им произволни имиња на променливите и користи ги како вредности во функцијата за прикажување на екран.

Активност 3: Обиди се да напишеш програмски код кој ќе го даде следниот излез:

Zbirot na broevite __ | __ e __

Razlikata na broevite __ | __ e __

Proizvodot na broevite __ | __ e __

Наставни содржини од ТЕМА 5: ДОПОЛНИТЕЛНИ СПЕЦИФИКИ НА ЈАЗИКОТ

Искази (техники) за внесување податоци во програмата и за објаснување податоци кои се очекуваат од корисникот

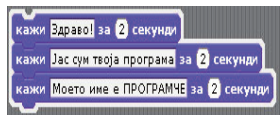
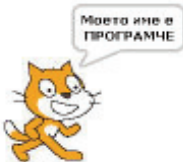
Цел на часот:

- ученикот/ученичката да ги осознае техниките за внесување на податоци во програмата;
- да ја осознае техниката за објаснувања за податоците кои се очекуваат од корисникот.

Воведен дел: Дискусија за интерактивни и не-интерактивни компјутерски програми – програми кои само даваат информации и програми кои бараат и издаваат информации до корисникот.

Главен дел: Запознавање со операциите во текстуалниот објект за внесување текстуални и бројни податоци преку пишување анализа и промени во пример програма.

Активност 1: Внеси ја пример програмата.


Програмски јазик Скреч	Програмски јазик Small basic
 	<pre>TextWindow.WriteLine („Zdravo“) TextWindow.WriteLine („Jas sum tvoja programa“) TextWindow.WriteLine („Moeto ime e PRO- GRAMCE.“)</pre>

Дали приказот на програмата се менува при секое ново извршување? Дали корисникот (а не програмерот) може да влијае или да го менува приказот од програмата?

Воочи: Програмите кои прикажуваат излез како непроменлива и независна информација од корисникот се неинтерактивни компјутерски програми.

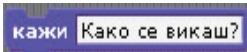
Интерактивните компјутерски програми овозможуваат внесување податоци од корисникот, работат со податоците и прикажуваат информации кои се променливи и зависат од влезните податоци на корисникот кои може да се менуваат при секое ново извршување на програмата.

Активност 2: Направи програма која ќе му се обрати на корисникот со неговото име.

Програмски јазик Скреч	Програмски јазик Small basic
 <p>Се објаснуваат исказите од категоријата “Распознавање” – ask...and wait (исказ кој чека влез од корисникот) и answer (исказ кој го чува одговорот)</p>	<p>како прва линија во програмскиот код внеси го исказот</p> <p>ime=TextWindow.Read (), а втората линија промени ја во TextWindow.WriteLine („Zdravo „+ ime)</p>

Изврши ја програмата. Исказот за внесување чека да внесеш име. По внесување на името ја испишува пораката со внесеното име. Ако корисникот не е програмерот кој ја пишувал програмата дали ќе знае што треба да напише или направи?

За да биде програмата разбирлива за корисникот, додади ја следната линија како прва во програмскиот код:

Програмски јазик Скреч	Програмски јазик Small basic
	<pre>WriteLine („Kako se vikas?“) Ако по оваа линија напишеме исказ за бришење на прозоречот како ќе се промени излезот на програмата? Провери</pre>

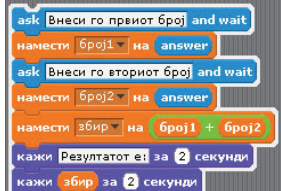

Изврши ја програмата!

Активност 3: Внеси ја и изврши ја пример програмата

Програмски јазик Скреч	Програмски јазик Small basic
 	<pre> broj1=2 broj2=3 zbir= broj1+broj2 TextWindow.WriteLine („2+3=“) TextWindow.WriteLine (zbir) </pre>

Дали програмата е интерактивна? Дали програмата бара внес на податоци од корисникот? За кои бројни вредности пресметува збир програмата?

Активност 4: Во пример програмата направи промени кои ќе му овозможат на корисникот внесување броеви по избор при секоје извршување на програмата.

Програмски јазик Скреч	Програмски јазик Small basic
 	<pre> TextWindow.WriteLine („Vnesi go prviot broj:“) broj1=TextWindow.ReadNumber () TextWindow.WriteLine („Vnesi go vtoriot broj:“) broj2=TextWindow.ReadNumber () zbir= broj1+broj2 TextWindow.WriteLine (broj1 + „+“ + broj2 + „=“ + zbir) </pre> <p>Воочи: Операција ReadNumber () му овозможува на корисникот внесување бројна вредност во програмата. Која операција е нова? Што овозможува во програмата?</p>

Завршен дел: Самостојна изработка на програма со зададен излез за практична примена на операцијата за внес на бројни податоци во програмата. Дискусија и насоки за работа.

Пример во **Small basic**

Активност 5: Направи нови промени во пример програмата за да го добиеш следниот излез:

Programa za sobiranje na tri broja

```

_____
Broj1=___
Broj2=___
Broj3=___
___+___+___=___
_____

```

Изработка на програми. Нови специфики на програмскиот јазик

Цел на часот:

ученикот/ученичката да осозне и да се оспособи за користење дополнителни специфики на јазикот; да се оспособи за изработка на едноставни програми.

Програмски јазик Скреч

А. Фигури во програмски јазик Скреч

Воведен дел

Која е основната фигура во Скреч (која фигура прва се појавува при стартување на програмскиот јазик)? Дали може да се менува, брише, зголемува или намалува?

Главен дел

Наставникот презентира пример програми за запознавање со другите фигури и можностите кои ги има програмскиот јазик за работа со фигури. Демонстрира како во Скреч се врши:

1.



2. Дуплирање, бришење, зголемување, намалување фигури со помош на алатките. По-мошно мени кое се појавува при десен клик врз некоја фигура од листата.



3. Менување на името на фигурата

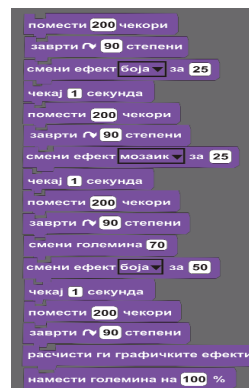
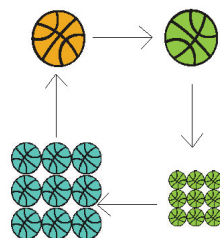


4. Искизи во Скреч за погледи (ефект) на фигура:

смени ефект боја за 25	Менување на визуелен ефект на фигурата за дефинирано време.
смени големина на 10	Поставување на визуелен ефект на дефиниран број
смени големина на 10	Менување на големина на фигурата
намести големина на 100 %	Поставување на големина на фигурата на дефиниран процент од оригиналната големина
расчисти ги графичките ефекти	Отстранување на сите графички ефекти на фигурата
покажи	Појавување на фигурата на Етапата
сокриј	Исчезување на фигурата од Етапата
оди напред	Поместување на фигурата пред сите фигури на Етапата
оди назад 1 нивоа	Поместување на фигурата наназад за дефиниран број на нивоа

Завршен дел:

Изработка на едноставни програми. Пример: Движење на топка во правец на стрелките на часовникот при што таа се трансформира.



Б. Маски во програмски јазик Скреч

Воведен дел

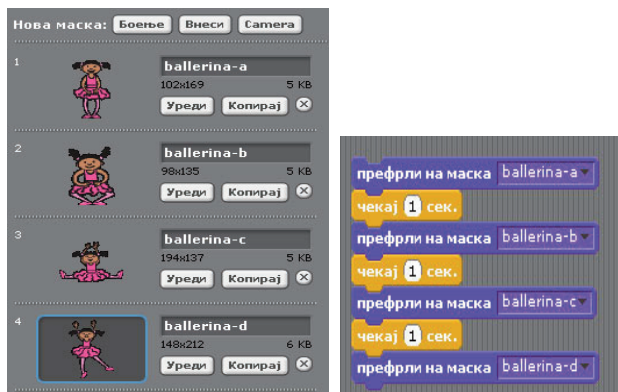
Презентација на програми изработени од претходниот час од страна на учениците и дискусија околу истите.

Главен дел

Наставникот го воведува и објаснува поимот маска.

Маските претставуваат различни облици на ист лик (фигура), во различни положби, бои, изрази... Изгледот на фигурата се менува со избор на јазичето "Маски".

Активност 1: Наставникот демонстрира едноставна програма за практична примена на маска на фигура. Пример:



Дискусија: Колку маски може да има една фигура? Дали може да се менуваат готовите маски? Дали може са,о да направиме маска? Дали е ограничен бројот на маски кој може да ги направиме?

Завршен дел

Изработка на едноставни програми од интерес на учениците со користење на маски.

В. Зелено знаме



Воведен дел

Наставникот презентира пример програма во кои има истовремена акција на повеќе фигури. Разговор: Колку фигури има во презентираната програма? Дали програмата се однесуваше само на една фигура или и другите фигури извршуваа некаква акција? Дали со досега научените техники постои начин истовремено да се активираат скриптите поврзани со различни фигури?



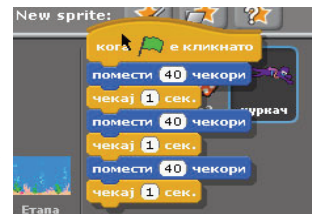
Главен дел:

Наставникот ја објаснува функцијата на копчето со зелено знаме.

Со притискање на копчето со зелено знаме се активира извршувањето на скриптата (програмата) на чиј врв се наоѓа блокот „кога зелено знаме е кликнато“. Зеленото знаме овозможува стартување на повеќе скрипти (програми) истовремено. Со притискање на копчето Enter се постигнува истиот ефект како и при притискање на зеленото знаме.

Учениците прават едноставна програма со истовремено движење на повеќе фигури

Забелешка: Една скрипта (програма) може да се ископира и да се додели на повеќе фигури. За таа цел се клика на програмата, се влече до новата фигура и се пушта.



Завршен дел

Учениците ги презентираат своите проекти. Учениците се самооценуваат според критериумите зададени од наставникот.

Г. Останати специфики во програмскиот јазик Скреч

Воведен дел

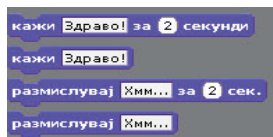
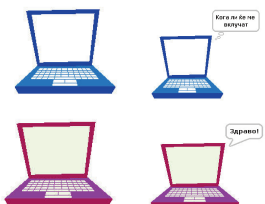
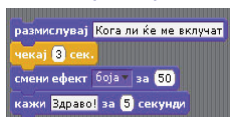
Дискусија – со кои можности на програмата Скреч се запознаа учениците?

Главен дел

1. Искази за дијалог

Наставникот ги објаснува останатите искази за дијалог на фигура и демонстрира градење програми со нивно употреба.

Пример:

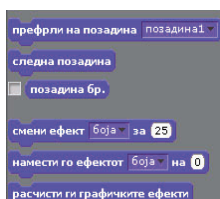


2. Менување етапа (позадина) во програмски јазик Скреч

Прашање:

Каква е етапата во нов Скреч проект?

Наставникот објаснува, демонстрира работа со етапа: јазиче „Позадина“, избор од Скреч библиотеката и самостојно креирање позадина, блокови (искази) за позадина.

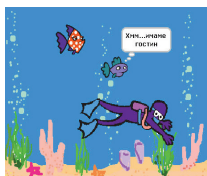
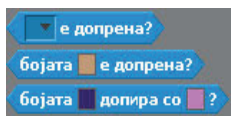


3. Искази (блокови) за распознавање

Наставникот објаснува и демонстрира работа со блокови за распознавање.

Завршен дел

Учениците креираат програми практично применувајќи го стекнатото знаење.



Програмски јазик Small basic

Воведен дел

Повторување на изучените објекти, операции и својства и дискусија за нивно комбинирано користење во практична изработка на нови програми.

Главен дел

Демонстрирање или испишување на готови пример програми за воведување нови објекти, операции или својства (на пр., нови објекти: часовник и математика и операции во нив).

Задавање проблемски ситуации за самостојна изработка на програми (пресметка на математички изрази, конвертор на валути или единици мерки, пресметувач на години и сл.) Излезот на програмата може да биде зададен или да се реализира според слободната замисла на ученикот.

Активност 1: Внеси ја пример програмата:

```
den= Clock .Day
mesec= Clock .Month
godina= Clock .Year
TextWindow .WriteLine („Denes e „+den+“-
“+mesec+“-“+godina+“ god.”)
```

или програмата со ист излез од една линија:
TextWindow .WriteLine („Denes e „+Clock .Date +“ god“)

Објектот Clock овозможува пристап и користење на системскиот часовник преку операциите (година, ден, час, датум и сл.

Активност 2: Внеси ја пример програмата:

```
TextWindow .Write („Vnesi broj:“)
broj = TextWindow .ReadNumber ()
koren= Math .SquareRoot (broj)
stepen= Math .Power (broj, 2)
TextWindow .WriteLine („Kvadraten koren od „+broj+“ e „+koren)
TextWindow .WriteLine („Vtori stepen od „+broj+“ e „+stepen)
```

Воочи: Објектот Math овозможува користење и пресметки со математички операции (корен, степен, остаток, случаен број).

Активност 3: Со примена на познатите и новите објекти, операции и својства обиди се да направиш:

1. програма која претвора внесен број на часови и минути во секунди;

2. програма која го прашува корисникот за годината на раѓање и му враќа информација за годините кои ги има (користи го објектот Clock, годините да се внесуваат во истата линија со прашањето, прашањето нека биде со црвена боја а информацијата за годините со жолта боја на текстот). На пр.

```
TextWindow.ForegroundColor="red"  
TextWindow.Write („Koja godina si roden?“)  
zbor= TextWindow.Read ()  
god= Clock .Year  
TextWindow.ForegroundColor="yellow"  
TextWindow.WriteLine („imas „+(god-zbor)+“ go-  
dini“)  
TextWindow.ForegroundColor="white"
```

3. програма која бара од корисникот да внесе вредност на катетите на еден правоаголен триаголник и потоа ја пресметува хипотенузата. (Програмата нека има подцртан наслов "Presmetuvawe hipotenuza na pravoagolen triagolnik", користи го објектот Math).

Завршен дел: Презентирање на програмите изработени од учениците пред останатите ученици. Споредба на различни изработки, анализа, дискусија.

Наставни содржини од ТЕМА 6: СТРУКТУРА ЗА ИЗБОР ОД ДВЕ МОЖНОСТИ

Структура и функционирање на блоковите за услов

Цел на часот:

– ученикот/ученичката да ја осознае синтаксата и семантиката на структурата за избор од две можности;

– да ја применува структурата за избор од две можности во конструкција на програми;

– да се запознае со концептот за блок од искази (наредби).

Воведен дел

Дискусија

1. Како би изгледал алгоритмот од Тема1-Телефонирање, ако линијата е зафатена?

2. Дискусија за проблемски ситуации од секојдневниот живот во кои е потребно проверување на исполнетост на одреден услов од што ќе зависи натамошниот тек на решавањето на проблемската ситуација.

Условот може да биде вистина или невестина. Ако условот е исполнет се извршува некоја активност. Ако не е исполнет, активността не се извршува, а може или немора да се изврши некоја друга активност.

Пр. „Ако светлото на семафорот е зелено помини ја улицата.“

„Ако цената на блузата е помала од 500 ден. купи ја блузата.

„Ако имам лист хартија и молив ќе ти напишам порака. Инаку ќе ти пратам меил.“

„Ако имам велосипед или ролерки ќе одам во парк. Инаку ќе останам дома.“

Ако речениците се напатствија за однесување, дали е задолжително да ја читаме до крај реченицата за правилно да го извршиме упатството, ако условот запишан на почетокот од реченицата не е исполнет? Објасни.

Главен дел

Наставникот ги запознава учениците со структурата за избор од две можности во соодветниот програмски јазик и презентира пример програми за запознавање со начинот на прикажување условни ситуации во програма, примената на условните блокови, нивната структура и функционирање.

Програмски јазик Скреч	Програмски јазик Small basic	Значење
	If услов Then искази Endif	Ако условот е исполнет се извршува исказот, а ако не е не се прави ништо друго.
	If услов Then искази Else искази Endif	Ако условот е исполнет се извршува одреден исказ, а ако не е исполнет се извршува друг исказ.

Активност 1: Анализирај алгоритам за проблемска ситуација во која се проверува точноста на зададен услов и се испишува соодветна порака на екран. Пишување програма според алгоритамот:

Програмски јазик Скреч	Програмски јазик Small basic
	TextWindow .Write („Vnesi lozinka:“) lozinka= TextWindow .Read () If lozinka=“JB007“ Then TextWindow .WriteLine („Vlezot e sloboden!“) EndIf

Дискусија: Дали ги препознаваш исказите кои се однесуваат на проверка на услов и донесување одлука?

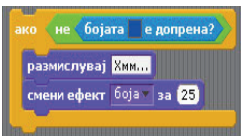
Исказите заедно се нарекуваат условен блок.

Каде почнува, а каде завршува условниот блок? Дали го препознаваш условот кој се проверува во условниот блок? Колку искази се напишани внатре во условниот блок? Дали исказот во условниот блок ќе се „помине“ ако условот не е исполнет?

Во програмските јазици проблемските ситуации во кои е потребно тестирање на услов од кои зависи текот на програмата се претставуваат со таканаречени условни блокови (или сложени искази за услов). Условните блокови се прикажуваат во неколку линии и се состојат од клучни зборови, услов, и искази и во нивната структура се разликува почеток, крај и внатрешни и скази. Градбата на блокот и клучните зборови се разликуваат од еден до друг програмски јазик но сепак се слични и нивната смисла е иста. На пример, вообичаено клучни зборови во блокот за услов се: *if, then, else, endif.*

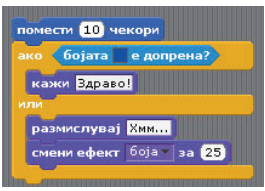
Условот може да биде запишан со употреба на споредбени или логички оператори. Најчести споредбени оператори се: = (кој е и оператор за доделување), <, >, <=, >=, <> (различно).

Активност 2: Додади во програмата уште еден условен блок

Програмски јазик Скреч	Програмски јазик Small basic
	If lozinka<>“JB007“ Then TextWindow .WriteLine („Lozinkata ne e točna!“) TextWindow .WriteLine („Vlezot e zabranet!“) EndIf

Дискусија: Што мислиш дали бројот на условните блокови во програмата е ограничен? Анализирај ги условите во условните блокови. Што можеш да заклучиш какви се меѓу себе?

Активност 3: Истата програма може да се запише на следниот начин

Програмски јазик Скреч	Програмски јазик Small basic
	TextWindow .Write („Vnesi lozinka:“) lozinka= TextWindow .Read () If lozinka=“JB007“ Then TextWindow .WriteLine („Vlezot e sloboden!“) Else TextWindow .WriteLine („Lozinkata ne e točna!“) TextWindow .WriteLine („Vlezot e zabranet!“) EndIf

Колку условни искази се напишани внатре во условниот блок?

Завршен дел

Активност 4:

Анализирај го начинот на кој се запишани деловите од условниот блок во пример програмата.

Како се запишани внатрешните искази во блокот во однос на линијата за почеток и крај на блокот? Како се запишани исказите во блокот еден во однос

на друг? Како се порамнети линиите за почеток и крај на блокот?

Со правилно порамнување на исказите во блокот се избегнуваат грешки, програмскиот код полесно се следи и разбира.

Забелешка: Во програмскиот јазик Скреч клучни зборови за услов се и зборовите “кога” и “додека” со соодветни блокови. Нивното значење и функционирање ќе биде прикажано подоцна.



Логички изрази како услов во условните блокови

Цел на часот:

- ученикот/ученичката да го осознае концептот на споредбен израз ($a > b$, $a = b$, ...);
- да се оспособи за конструирање едноставни споредбени изрази (услови);
- да конструира логички изрази (сложени услови) преку конјункција, дисјункција и негација на споредбени изрази (услови);
- да ја применува структурата за избор од две можности во конструкција на програми.

Воведен дел: Дискусија за ситуации од секојдневниот живот со логички услови.

Кога следните искази ќе бидат вистина а кога лага?

Ако имам пари **И** ако продавницата е отворена ќе купам портокали.

Ако одам на пазар **ИЛИ** ако одам во продавница ќе купам портокали.

– Обиди се самиот да дадеш пример од секојдневниот живот.

– Како би го запишале условот: бројот да биде во опсегот од 30 до 50, а како условот бојот да биде надвор од опсегот од 30 до 50?

Главен дел: Запознавање со логичките оператори и буловата логика. Изработка на програма според понуден графички алгоритам со условен блок и логички услов. Запознавање со исказите за внесување звук во извршниот дел од програмата.

Воочи: Најчести логички оператори при запишување на услов се: AND(И) и OR(ИЛИ). Логичките оператори функционираат според правилата на таканаречената Булова логика според која ако изразот го разбереме како „лева страна ОПЕРАТОР десна страна“ тогаш:

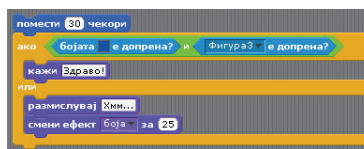
За оператор AND (И):			За оператор OR (ИЛИ):		
Лева страна	Десна страна	Конечен одговор	Лева страна	Десна страна	Конечен одговор
Вистина	Вистина	Вистина	Вистина	Вистина	Вистина
Вистина	Лага	Лага	Вистина	Лага	Вистина
Лага	Вистина	Лага	Лага	Вистина	Вистина
Лага	Лага	Лага	Лага	Лага	Лага

– Обиди се ситуациите од табелите да ги претставиш со графички алгоритам.

Активност 1: Анализа на графички алгоритам и пишување програма која користи логички оператори во условот.

Пример:

Програмски јазик Скреч



```

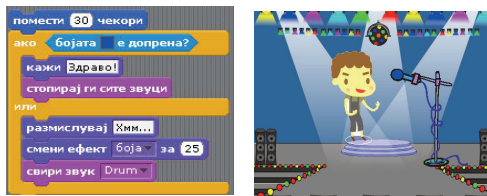
Програмски јазик Small basic
TextWindow .Write („Vnesi korisnicko ime:")
ime = TextWindow .Read ()
TextWindow .Write („Vnesi lozinka:")
lozinka = TextWindow .Read ()
If ime="monster" AND lozinka="JB007" Then
TextWindow .WriteLine („Vlezot e sloboden")
EndIf
    
```

Замени го логичкиот оператор со OR (ИЛИ) и анализирај го излезот.

Завршен дел

Изработка на едноставни програми со условни наредби и примена на звук. Пример:

Програмски јазик Скреч



Програмски јазик Small basic

Внеси промени во програмата за пораката "Vlezot e sloboden" да се испише на екран со зелена позадина, а пораката "Vlezot e zabranet" на екран со црвена позадина.

Можно решение:

```
TextWindow .Write („Vnesi lozinka:“)
lozinka= TextWindow .Read ()
If lozinka="JB007" Then
TextWindow . ForegroundColor ="green"
TextWindow .WriteLine („Vlezot e sloboden!“)
Sound . PlayChime ()
Else
TextWindow . BackgroundColor ="red"
TextWindow .WriteLine („Lozinkata ne e točna!“)
TextWindow .WriteLine („Vlezot e zabranet!“)
Sound . PlayBellRing ()
EndIf
```

Техника на вгнездување искази

Цел на часот

– ученикот/ученичката да ја осознае техниката на вгнездување на искази и резултатите од истото при извршувањето на програмите;

– да ја применува техниката на вгнездување на искази во конструкција на програми.

Воведен дел

Дискусија за проблемска ситуација од секојдневниот живот во која еден условен настан зависи од друг условен настан. Едниот услов е надворешен, а другиот внатрешен – вгнезден. До вгнездениот услов

„се стигнува“ само ако надворешниот услов е исполнет. Пример (псевдокод):

Ако имам пари и одам во продавница тогаш

Ако има портокали тогаш ќе купам 2 кила

крај

Ако бананите се зрели тогаш

ќе купам 3 кила

инаку

ќе купам 1 кило

крај

Крај

Кој услов е надворешен, а кои услови се внатрешни (вгнездени)?

Главен дел

Анализа на графички алгоритам за проблемски ситуации кои се претставуваат со вгнездени услови. Пишување програма според алгоритмот и запознавање со терминот вгнездени условни блокови и начин на нивно правилно запишување и користење.



Активност 1:

Претстави ја со графички алгоритам следната проблемска ситуација:

За внесен број од корисникот провери дали припаѓа на опсегот од 50 до 100 и ако е во тој опсег проверува дали е парен или непарен и испишува порака до корисникот.

Анализа на алгоритмот.

Активност 2: Напиши програма според алгоритмот.

Програмски јазик Скреч	Програмски јазик Small basic
 <p>ask Внеси број во опсег од 50 до 100 and wait</p> <p>ако answer > 49 и answer < 101</p> <p>ако answer модул 2 = 0</p> <p>кажи Бројот е парен</p> <p>или</p> <p>кажи Бројот е непарен</p> <p>или</p> <p>ако length of answer = 0</p> <p>кажи Не е внесен број</p> <p>или</p> <p>кажи Бројот не припаѓа во бараниот опсег</p> 	<p>Во програмите се користи нова операција Remainder на објект Math за пресметување остаток при делење на два броја. Еден можен запис на програмскиот код</p> <pre> TextWindow Write („Vnesi broj:“) broj= TextWindow .Read () If broj >=50 And broj<=100 Then ostatok= Math . Remainder (broj,2) If ostatok=0 Then TextWindow .WriteLine („Brojot e paren!“) EndIf If ostatok<>0 Then TextWindow .WriteLine („Brojot e neparen!“) EndIf EndIf </pre>


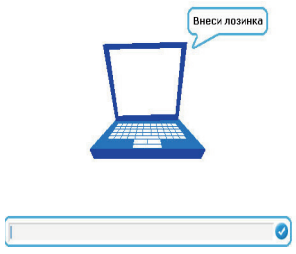
Дали ќе се поминат вгнездените услови ако надворешниот услов не е исполнет?

Исказ во условен блок може да биде друг условен блок. Ако еден условен блок се наоѓа запишан внатре во друг условен блок како исказ тогаш условните блокови се вгнездени. Извршувањето на вгнездениот условен блок зависи од исполнувањето на условот во условниот блок во кој е вгнезден.

Завршен дел

Самостојно пишување програма според одредени барања и со давање насоки и сугестии за работа со примена на условни блокови.

Активност 3. Напиши програма која од корисникот бара да внесе лозинка и му дозволува да внесе корисничко име само ако лозинката е правилно запишана. Ако и корисничкото име е правилно запишано му дозволува влез во спротивно влезот не му е дозволен. Можно решение:

Програмски јазик Скреч	Програмски јазик Small basic
 <p>ask Внеси лозинка and wait</p> <p>ако answer = JB007</p> <p>ask Внеси корисничко име and wait</p> <p>ако answer = monster</p> <p>кажи Влезот е слободен</p> <p>или</p> <p>кажи Влезот е забранет</p> <p>или</p> <p>кажи Лозинката е погрешна</p> 	<pre> TextWindow .Write („Vnesi lozinka:“) lozinka= TextWindow .Read () If lozinka="JB007" Then TextWindow .WriteLine („Vnesi korisnicko ime“) ime=TextWindow .Read () If ime="monster" Then TextWindow .WriteLine („Vlezot e sloboden!“) EndIf Else extWindow .Writeln („Vlezot e zabranet!“) EndIf </pre>

Наставни содржини од ТЕМА 7: СТРУКТУРА ЗА ИЗБОР ОД ПОВЕЌЕ МОЖНОСТИ

Структура за избор од повеќе можности

Цел на часот:

– ученикот/ученичката да ја осознае синтаксата и семантиката на структурата за избор од повеќе можности;

– да ја применува структурата за избор од повеќе можности во конструкција на програми.

Воведен дел

Дискусија за спротивставени услови и алтернативни услови.

Ако времето е сончево ќе одам на прошетка, а ако не е не одам на прошетка.

Ако времето е сончево ќе одам на прошетка, а ако времето е облачно ќе одам кај другарот, а ако врне ќе спијам.

Обиди се да дадеш сличен пример. Во компјутерските игри многу често корисниците избираат една од понудените можности од менито на програмата за број на играчи, ниво, почеток или крај и сл. Од изборот зависи натамошниот тек во програмата.

Во досегашните примери на програми со условни блокови условот е или исполнет или неисполнет. Ако е исполнет се изведува одредена активност, а ако не е не се прави ништо или се изведува некоја друга активност. Условите беа спротивставени. Има и други проблемски ситуации со алтернативни услови при кои ако еден услов не е исполнет се проверуваат алтернативни услови и во зависност од тоа кој е исполнет се изведува некоја активност.

Главен дел

Анализа и дискусија по пример програма за запознавање со структурата на блокот со алтернативни услови.

Активност 1: Напиши ја програмата: Колку години имаш? Ако имаш помалку од 13 години тогаш треба да се испише пораката “Ти си дете”. Ако имаш помалку од 20 години тогаш треба да се испише пораката “Ти си тинејџер”. Ако имаш помалку од 100

години тогаш треба да се испише пораката “Ти си возрасен човек”.

Програмски јазик Скреч	Програмски јазик Small basic
 <p>Scratch code block: ask "Колку години имаш?" and wait. Then a series of if-then blocks: if answer < 6 then say "Ти си дете"; if answer < 20 then say "Ти си тинејџер"; if answer < 100 then say "Ти си возрасен човек".</p>	<p>Запознавање со нова операција Round (број) од објектот Math која децималниот број го заокружува на цел.</p> <pre> TextWindow .Write („Kolku godini imas?“) god = TextWindow .ReadNumber () If god<13 Then TextWindow . WriteLine („Ti si dete!“) Elseif god<20 then TextWindow .WriteLine („Ti si ti- nejdzer!“) Elseif god<100 then TextWindow .WriteLine („Ti si vo- zrasen covek!“) EndIf </pre>



Дали ако првиот услов е исполнет ќе се проверуваат следните? Дали може во овој конкретен пример условите да си ги променат местата? Зошто?

Воочи: Структурата на блокот за избор на алтернативни услови е следна:

Програмски јазик Скреч	Програмски јазик Small basic
 <p>Scratch code block: if then structure. if then block with 'или' (or) connector. if then block with 'или' connector. if then block with 'или' connector. if then block with 'или' connector.</p>	<pre> If услов1 Then Исказ1 Elseif услов2 Then Исказ 2 Elseif услов3 Then Исказ 3 EndIf </pre>

Анализа, дискусија и објаснување на пример програма за избор на една од повеќе понудени можности и различен тек на програмата за секој од изборите.

Активност 2.

Програмски јазик Скреч	Програмски јазик Small basic
<p>Како изгледа Марко кога</p> <ol style="list-style-type: none"> 1. Ужива во убав ден 2. Се радува на успехот на ракометниот тим за кој навива 3. Не сака да слуша поша музика <p>Како изгледа Марко кога</p> <ol style="list-style-type: none"> 1. Ужива во убав ден 2. Се радува на успехот на ракометниот тим за кој навива 3. Не сака да слуша поша музика  <p>Внеси број 1,2 или 3</p> 	<p>Напиши програма во која како излез ќе се прикаже како изгледа Марко кога</p> <ol style="list-style-type: none"> 1. ужива во убав ден 2. се радува на успехот на ракометниот тим за кој навива 3. не сака да слуша лоша музика. <p>Напиши програмата за претворање на денари во евра и евра во денари користејќи мени за избор.</p> <pre> TextWindow.WriteLine („Programa za pretvaranje na valuti") TextWindow.WriteLine („-----") TextWindow.WriteLine („1- Pretvaranje na denari vo evra") TextWindow.WriteLine („2- Pretvaranje na evra vo denari") TextWindow.WriteLine („3- Izlez") TextWindow.WriteLine („-----") TextWindow.Write („Vnesi broj: 1, 2 ili 3:„) br= TextWindow.ReadNumber () If br=1 Then TextWindow.Write („Vnesi denari") den= TextWindow.ReadNumber () evra=den/61.5 evra1=den/61.5 evra=Math.Round (evra1) TextWindow.WriteLine (den+„ denari se „+evra+“ evra“) Elseif br=2 Then TextWindow.WriteLine („Vnesi evra") evra= TextWindow.ReadNumber () den=evra*61.5 TextWindow.WriteLine (evra +„ evra se „+den +“ denari“) Elseif br=3 Then TextWindow.WriteLine („Kraj") EndIf </pre>

Дали условите во овој пример може да си ги променат местата? Зошто?

Завршен дел

Самостојно пишување програма според одредени барања и со давање насоки и сугестии за самостојна практична примена на блоковите со повеќе услови.

Активност 3:

Напиши програма која бара од корисникот да внесе број како одговори на прашањето "Kolku palacinki mozes da izedes?" и во зависност од одговорот да му испише три различни пораки: ≤ 5 "Slabo...treniraj! J", ≤ 10 "Ne loso..ke te bide!" и за друг одговор "Nemoj da preteruvash ke ti se slosi!"

или

Напиши програма со мени која од корисникот бара да внесе два броја а потоа од менито корисникот избира собирање, множење или одземање на броевите.

Наставни содржини од ТЕМА 8: ОСНОВНА СТРУКТУРА ЗА ПОВТОРУВАЊЕ

Сложен исказ за повторување – циклус со услов

Цел на часот:

– ученикот/ученичката да ја осознае синтаксата и семантиката на структурата за повторување на циклус до исполнување на услов;

– да се оспособи за примена на структурата за повторување на циклус до исполнување на услов во конструкција на програми.

Воведен дел: Дискусија за ситуации од секојдневниот живот во кои одредена активност се повторува # додека одреден услов е исполнет:

- додека има тесто прави палачинки;
- додека не добиеш слободно бирај го бројот;
- додека има листови печати копии.

Дали во овие ситуации однапред се знае бројот на палачинки, бројот на бирање на телефонскиот број или бројот на копии?

При пишувањето на програмите многу често се јавува потреба од повторување на еден ист исказ повеќе пати. За да не се повторува пишувањето на исказот, во програмскиот код се користи структура за повторување која се вика циклус, петелка, јамка или loop.

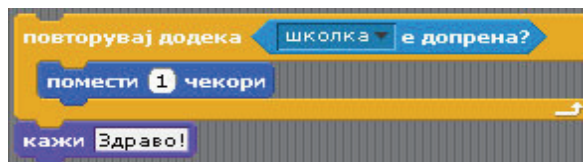
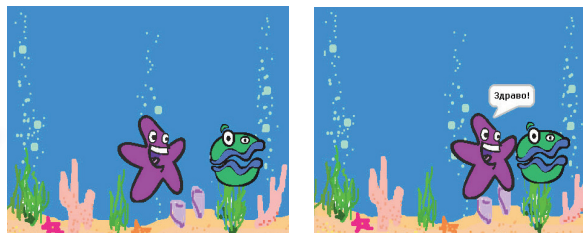
Главен дел: Во зависност од начинот на кој функционираат постојат неколку видови циклуси

(јамки). Запознавање со циклус со услов. Пишување пример програма со циклус со услов и преку нејзина анализа запознавање со структурата на циклус со услов.

Активност 1:

Програмски јазик Скреч

Напиши програма со која една фигура ќе се движи сè додека не допре друга фигура.



Програмски јазик Small basic

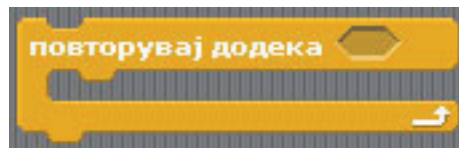
Напиши го програмскиот код кој од корисникот бара да ја внесува лозинката сè додека не ја внесе правилно:

```
While lozinka<>"JB007"  
TextWindow .Write („Vnesi lozinka:")  
lozinka= TextWindow .Read ()  
EndWhile  
TextWindow .WriteLine („Точна lozinka!")
```

Дали го препознаваш циклусот со услов? Кој е условот? Кога ќе се изврши исказот во циклусот?

Воочи: Структурата на циклус со услов е:

Програмски јазик Скреч



Циклусот се повторува сè додека условот се исполни

Програмски јазик Small basic

While услов

искаж

EndWhile

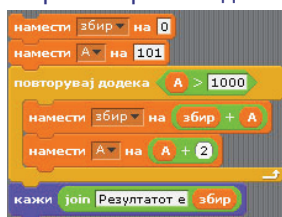
Циклусот се повторува сè додека условот е исполнет

Условот во циклусот може да биде споредбен или логички.

Активност 2:

Програмски јазик Скреч

Напиши програма за пресметување на збир на непарните броеви од 101 до 1000.



Програмски јазик Small basic

Напиши програма за внесување броеви. Броевите се внесуваат сè додека нивниот збир е помал или еднаков на 100.

```
zbir=0
```

```
While zbir<=100
```

```
TextWindow .Write („Vnesi broj:“)
```

```
broj = TextWindow .ReadNumber ()
```

```
zbir = zbir+broj
```

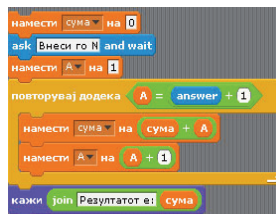
```
TextWindow .WriteLine („zbir="+zbir)
```

```
EndWhile
```

```
TextWindow .WriteLine („Zbirot na vnesenite broevi e pogolem od 100“)
```

Програмски јазик Скреч

Напиши програма за пресметување збир (сума) на N природни броеви, ако N=6, треба да се пресмета 1+2+3+4+5+6. Можмо решение:



Програмски јазик Small basic

Напиши програма која внесен број од корисникот го дели со два сè додека количникот е поголем од еден и ги испишува меѓу резултатите. на пример:

```
TextWindow .Write (“Vnesi broj:“)
```

```
broj= TextWindow .ReadNumber ()
```

```
While broj>1
```

```
broj/2
```

```
TextWindow .WriteLine (broj)
```

```
EndWhile
```

Дали евозможен случај кога циклусот ниту еднаш нема да биде поминат? Објасни.

Дали може да има бесконечен циклус? Размисли каков услов тогаш треба да биде поставен во циклусот.

Завршен дел: Самостојна изработка на програма со насоки и упатства за примена на новата структура.

Активност 3:

Пример:

Наставни содржини од ТЕМА 9: ОСТАНАТИ СТРУКТУРИ ЗА ПОВТОРУВАЊЕ

Сложен искаж за повторување – циклус со бројач

Цел на часот:

– ученикот/ученичката да ја осознае синтаксата и семантиката на структурата за повторување на циклус со броење на циклусите;

– да се оспособи за примена на структурата за повторување на циклус со броење на циклусите во конструкција на програмите.

Воведен дел

Дискусија за ситуации од секојдневниот живот во кои одредена активност се повторува точно определен број на пати кој е однапред познат.

- Направи 5 палачинки преку следните активности: земи тесто, истури го во тава, почекај 1 минута, преврти ја палачинката, почекај 1 минута, стави ја во чинија.

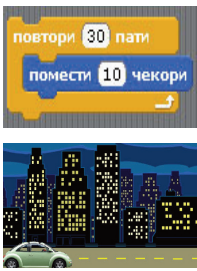
- Постави 4 места на масата преку следните активности: стави чинија, лажица, виљушка, нож и чаша.

Направи споредба со ситуациите во кои се повторува активната додека условот е исполнет. Дали овде има услов кој се проверува? Дали во овие ситуации однапред е познат бројот на повторување на активностите?

Главен дел Запознавање циклус со бројач. Циклусот со бројач се извршува дефиниран број пати. Пишување пример програма со циклус со бројач и преку нејзина анализа запознавање со структурата.

Активност 1:

Анализирај го излезот!

Програмски јазик Скреч	Програмски јазик Small basic
Напиши програма за движење на фигура со користење циклус со бројач. Пример	Напиши го пример програмскиот код со циклус со бројач кој го испишува зборот "Zdravo" 10 пати.
	For broj=1 To 10 TextWindow.WriteLine "Zdravo") EndFor

Активност 2:

Анализирај го излезот!

Воочи: Структурата на циклус (јамка) со бројач е:

Програмски јазик Скреч	Програмски јазик Small basic
	For бројач=вредност1 To вредност2 Step вредност исказ EndFor

Циклусот со бројач функционира на тој начин што при секое „вртење“ ја зголемува вредноста на бројачот за вредноста на чекорот почнувајќи од најмалата сè додека не ја достигне најголемата. Ако чекорот не е напишан се поставува на вредност еден.

Дали во првиот пример се користеше вредноста на бројачот во исказите во циклусот? Дали во вториот пример се користеше вредноста на бројачот во исказите во циклусот?

Воочи: Во циклус со бројач во зависност од проблемската ситуација вредноста на бројачот може и немора да се користи во исказите во јамката.

Активност 3: Програмски јазик Small basic

Промени ја втората програмата на следниот начин:

```
For broj=1 To 10 Step 2  
TextWindow.WriteLine (broj)  
EndFor
```

Анализирај го излезот на програмата. Промени го чекорот на -1. Дали ќе врти бројчот? Зошто? Дали е возможно петелката со бројач да не биде ниту еднаш помината? Објасни.

Како треба да го промениш опсегот на бројачот? Анализирај го излезот.

Самостојна изработка на едноставни програми за практично користење на циклусот со бројач при пишување на програмски код.

Активност 4:

Програмски јазик Small basic

Напиши програма која од корисникот бара да внесе број, а потоа испишува таблица на множење од 1 до 10 со тој број:

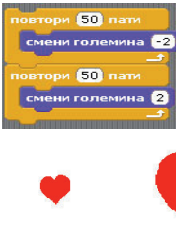

```

TextWindow .Write („Vnesi broj:“)
broj= TextWindow .Read ()
TextWindow .WriteLine („Tablica na mnozenje od 1
do 10 so brojot „+broj)
For br=1 To 10
proizvod=broj*br
TextWindow .WriteLine (br+“*“+broj+“=“+proiz-
vod)
EndFor

```

Дали во исказите во јамките се користи вредноста на бројачот? Внеси промени во програмата за опсегот на броеви со кои се множи внесениот број, множењето да биде за парни или непарни броеви и сл.

Активност 5:

Програмски јазик Скреч	Програмски јазик Small basic
Напиши програма со која ќе се намалува и зголемува големината на една фигура 50 пати	Напиши го пример програмскиот код со циклус кој ги испишува броевите од 1 до 10.
	<pre> For broj=1 To 10 step 1 TextWindow .WriteLine (broj) EndFor </pre>

Напиши една интересна програма во вид на анимација која овозможува со држење притиснато кое било копче од тастатурата лансирање на ракета.

```

TextWindow .ForegroundColor =“red“
TextWindow .WriteLine („Drzi go kopceto SPACE i
lansiraj ja raketata“)
TextWindow .Pause ()
TextWindow .ForegroundColor =“green“
For broj_na_red=20 To 0 Step 1
TextWindow .CursorTop =broj_na_red
TextWindow .WriteLine („/\“)
TextWindow .WriteLine („| |“)

```

```

TextWindow . WriteLine („| |“)
TextWindow . WriteLine („—“)
TextWindow . WriteLine („|\“)
TextWindow . PauseWithoutMessage ()
TextWindow .Clear ()
EndFor

```

Анализа и дискусија по програмскиот код и излезот. Дали разбра како е направена анимацијата? Зошто е потребна првата линија со пауза надвор од циклусот? Што се постигнува со линијата со пауза внатре во циклусот?

Воочи ја новата операција во текстуалниот објект `PauseWithoutMessage ()` која без порака до корисникот чека корисникот да притисне било кое копче од тастатурата за да продолжи со извршување програмскиот код.

Воочи го новото својство на текстуалниот прозорец `CursorTop` со кое се позиционира курсорот во одреден ред од прозорецот.

Завршен дел

Изработка на програми од интерес на учениците.

Дополнителни структури за повторување во Скреч

Цел на часот:

– ученикот/ученичката да ја осознае синтаксата и семантиката на друга структура за повторување.

Воведен дел

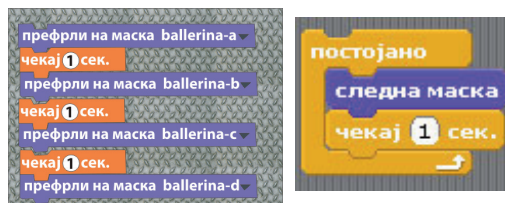
Дискусија за ситуации од секојдневниот живот во кои одредена активност се повторува постојано (на пр. изгрејсонце и зајдисонце). Дали овде има услов кој се проверува?

Главен дел: Запознавање со дополнителни структури (блокови) за повторување во Скреч. Демонстрација на пример програма со циклус во која постојано се повторуваат исказите кои се во циклусот и преку нејзина анализа запознавање со структурата.

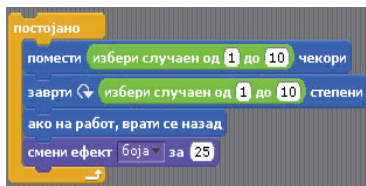
Воочи:

Програмски јазик Скреч	
Блок (структура)	Значење
	Постојано извршување на блоковите (исказите) кои се внатре во циклусот.

Активност 1: Разгледај го повторно примерот со балерината која менува маски. Дали може фигурата-балерина постојано да танцува? Како? Анализирај го излезот.



Активност 2: Програма со која текстот “информатика” произволно ќе се движи и притоа постојано ќе ја менува бојата.



Дискусија околу активности од секојдневниот живот кои постојано се повторуваат, но се условени.

На пример: Јана секој ден оди во училиште. Дали оваа активност е условена? Дали ќе се изведе ако Јана е болна, ако е неработен ден и сл.?

Демонстрација на пример програма со циклус со услов во која постојано се повторуваат исказите кои се во циклусот и преку нејзина анализа запознавање со структурата.

Воочи:

Програмски јазик Скреч	
Блок (структура)	Значење
	Постојано извршување на блоковите (исказите) внатре ако поставениот услов е точен.

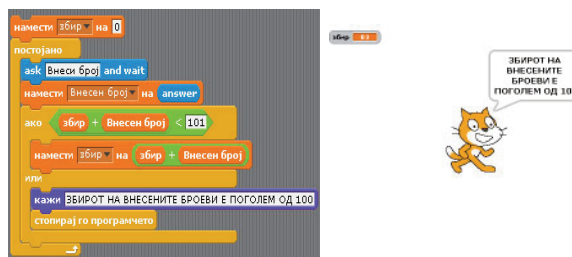
Активност 3: Програма со која ако фигурата-дух се допре со покажувачот тогаш се менува неговата боја.



Дали знаеме колку пати ќе се повторува циклусот? Кој е условот? Кога ќе се изврши исказот во циклусот?

Завршен дел

Активност 4: Напиши програма за внесување броеви. Броевите се внесуваат сè додека нивниот збир е помал или еднаков на 100. Можно решение:



Изработка на програми

Цел на часот:

– ученикот/ученичката да се оспособи за изработка на програми со досега изучените техники.

Воведен дел

Повторување: Што е тоа циклус со бројач? По што се разликува од циклусот со услов? Што е почетна, што е крајна вредност, а што чекор на менување кај бројачот? Дали чекорот може да биде негативен? Каков треба да биде тогаш опсегот? Дали вредноста на бројачот може да се користи во изрази во бројачот? А дали мора да се користи? Дали постојат циклуси кои се извршуваат неограничен број пати?

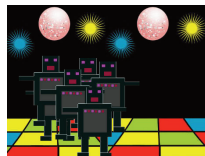
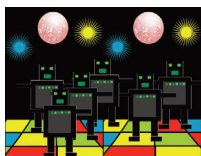
Главен дел

Пишување на интересни пример програми и анализа и самостојно пишување на едноставни програми со примена на циклус со бројач и досега изучените можности на програмскиот јазик.

Активност 1:

Програмски јазик Скреч

Напиши програма со која ќе овозможиш постојано танцување на фигурата-робот. Притоа постојано да свира мелодија и да се менува ефектот-боја на роботите. Нацртај сам позадина. Можеш да додадеш и дополнителни елементи. Забавувај се.



Програмски јазик Small basic

Напиши една интересна програма во вид на анимација која овозможува со клик на кое било копче од тастатурата “мрдање” на човечје.

```
TextWindow.WriteLine („Covece sto vezba“)
For i=1 To 5
TextWindow.WriteLine („<o>“)
TextWindow.WriteLine („ | „)
TextWindow.WriteLine („ ^ “)
```

```
TextWindow.PauseWithoutMessage ()
TextWindow.Clear ()
TextWindow.WriteLine („o“)
TextWindow.WriteLine („<|>“)
TextWindow.WriteLine („^“)
TextWindow.PauseWithoutMessage ()
TextWindow.Clear ()
TextWindow.WriteLine („ o“)
TextWindow.WriteLine („—|—“)
TextWindow.WriteLine („ ^ “)
TextWindow.PauseWithoutMessage ()
TextWindow.Clear ()
EndFor
TextWindow.ForegroundColor = “red“
TextWindow.WriteLine („Dosta bese! Se izmoriv!“)
Со кои искази се постигнува ефектот на анимација во излезот на програмата?
```

Активност 2:

Програмски јазик Скреч

Напиши програма со која ќе овозможиш постојано движење на рибите во аквариум во случаен правец.

Напиши програма со која ќе прикажеш сцена од некоја бајка.

Напиши програма со која ќе симулираш сообраќајница.

Напиши програма која ги собира парните броеви во опсегот од 10 до 20.

Програмски јазик Small basic

Напиши програма која 1000 пати го испишува името на корисникот. Од лево до десно и од горе до доле го полни цел екран со зборот! (користи Write).

Напиши програма која назименично пишува два збора внесени од корисникот во различна боја одреден број пати.

Напиши програма која ги испишува парните броеви во опсегот од 10 до 20.

Напиши програма која ги испишува непарните броеви броевите од 19 до 1.

Завршен дел: Анализа и дискусија по напишаните програми.

Вгнездени циклуси

Цел на часот:

- ученикот/ученичката да ја осознае синтаксата и семантиката на структурата за вгнездени јамки;
- да се оспособи за примена на структурата за вгнездени јамки во конструкција на програмите.

Воведен дел

Дискусија за ситуации од секојдневниот живот во кои одредена активност се повторува точно определен број пати и активноста се состои од повторување одреден број пати на друга активност:

На пример: Во 5 чаши стави една по една 10 цреши.

На тројца играчи подели им по шест карти.

Бројот на повторувања е познат и во двете поврзани ситуации. Пример (со псевдојазик)

Земи чаша

Чаши = 1 до 5

Цреши = 1 до 10

Земи цреша


Стави во чаша

Следна цреша

Следна чаша

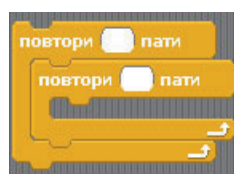
Главен дел: Пишување и анализа на пример со примена на вгнездени циклуси.

Активност 1: Напиши го следниот пример програмски код и анализирај го приказот:

Програмски јазик Скреч	Програмски јазик Small basic
	<pre>For broj1=1 To 5 TextWindow .WriteLine (broj1) For broj2=1 To 3 TextWindow .WriteLine („ZDRAVO“) EndFor EndFor</pre>

Што прикажува излезот на програмата? Кој циклус е внатрешен, а кој надворешен? Кој циклус „врти“ прв? Дали вредноста на бројачот од надворешниот циклус се користи во внатрешниот циклус?

Воочи: Изглед и начин на запишување на структурата за вгнездени циклуси

Програмски јазик Скреч	Програмски јазик Small basic
	<pre>For бројач 1=вредност To вредност Step вредност For бројач 2=вредност To вредност Step вредност исказ EndFor EndFor</pre>

Активност 2: Во програмата промени ја крајната вредност за бројачите, зборот што се испишува, додади можеби линија за прашање до корисникот за збор, број на пати на пишување на зборот, боја на позадина или текст за да го промениш излезот на програмата на интересен начин.

Активност 3:

Програмски јазик Скреч	Програмски јазик Small basic
<p>Направи програма со која континуирано ќе се движи некоја фигура и притоа ќе менува маски.</p>	<p>Во програмата за лансирање ракета додади надворешен јамка кој врти 5 пати околу постоечката јамка и анализирај го излезот. Објасни.</p>
	<pre>TextWindow .ForegroundColor =“red“ TextWindow .WriteLine („Drzi go kop- ceto SPACE i lansiraj ja raketata“) TextWindow .Pause () TextWindow .ForegroundColor =“green“ For lansiranja=1 To 5 For brojnared=20 To 0 Step -1 TextWindow .CursorTop =brojnared TextWindow .WriteLine („^“) TextWindow .WriteLine („ “) TextWindow .WriteLine („—“) TextWindow .WriteLine („/\,“) TextWindow .PauseWithoutMessage () TextWindow .Clear () EndFor EndFor</pre>

Дали вредноста на бројачот од надворешниот циклус се користи во внатрешниот циклус?

Завршен дел

Програмски јазик Скреч

Дискусија, анализа, упатства за работа и самостојно креирање програми од интерес на учениците.

Програмски јазик Small basic

Дискусија: Во претходните пример програми на вгнездени циклуси вредноста на надворешниот бројач не се користеше во изведување на активностите во внатрешниот бројач. Следната програма ја користи вредноста од надворешниот бројач за изведување на активностите во внатрешниот бројач.

Дискусија, анализа, упатства за работа и самостојно внесување промени во програмата за таблица на множење на внесен број со броевите од 1 до 10 во програма која ја испишува таблицата на множење до 10 за сите броеви од 1 до 10.

Активност 4. Едно можно решение:

TextWindow.WriteLine („Таблица на множење од 1 до 10“)

```
For br1=1 To 10
```

```
TextWindow.WriteLine („-----“)
```

```
For br2=1 To 10
```

```
  proizvod=br1*br2
```

```
TextWindow.WriteLine (br1+“*“+br2+“=”+proizvod)
```

```
EndFor
```

```
TextWindow.PauseWithoutMessage ()
```

```
EndFor
```

Како ја искористивме вредноста на бројачот од надворешниот циклус во внатрешниот циклус? Зошто користиме линија за пауза во надворешниот циклус?

Наставни содржини од ТЕМА 10: ПРИМЕНИ ВО ГРАФИЧКО ПРОГРАМИРАЊЕ И ЖЕЛКА-ГРАФИКА

Опис на околината за графички излез. Техники за приказ на графички елементи

Цел на часот:

– ученикот/ученичката да се запознае со концептите на програмирање на графика и можностите кои ги нуди ПЈ;

– да се оспособи да ги користи изучените техники за програмирање при изработка на различни графички прикази.

Програмски јазик Скреч

Воведен дел

Повторување за етапа како простор каде може да се гледаат приказните, игрите и анимациите и каде фигурите се движат и дејствуваат меѓусебно.

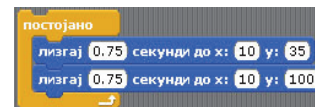
Главен дел

Запознавање со карактеристиките на Етапата. Широка е 480 единици, а висока 360 единици. Целата површина е поделена со координатна мрежа. Центарот на координатниот систем, точката (0,0)

се наоѓа точно во средината. Горниот лев агол има координати (-240, 180), а долниот десен агол (240, -180). Презентација на програми кои користат изрази со (x,y) координати и нивна анализа.

Активност 1:

Презентирање на пример програма со употреба на координати и анализа на истата. Пример: скокач на трамболина.



Зошто се потребни (x,y) координатите? Дали може да се направи истата програма без употреба на (x,y) координати?

Активност 2:

Презентирање на пример програма со употреба на координати и анализа. Пример: програма која една иста фигура поставена повеќе пати на Етапа се движи, т.е. го следи движењето на глумчето. Дали за сите фигури мора да се направи посебна скрипта? Во што е единствената разлика во скриптите на фигури-те?



Прикажана е скрипта (програма) за првата фигура. Останатите фигури имаат иста скрипта, но различни координати

Завршен дел

Изработка на програми со користење координати. Пример: Програма со рок група која свири.

Програмски јазик Small basic

Воведен дел

Повторување за текстуалниот прозорец (објектот TextWindow) како прозорец кој има одредени својства и во него се изведуваат одредени операции при што се прикажуваат текстуални и бројни податоци.

Прашања: Кои својства на текстуалниот прозорец ги применувавме во досегашните програми? Кои операции ги применувавме во текстуалниот прозорец? Какви податоци се прикажуваат во текстуалниот прозорец?

Запознавање со графичкиот прозорец како објект во програмскиот јазик во кој се цртаат точки, линии, текстови, различни геометриски фигури и слики. Запознавање со основните својства на графичкиот прозорец (име, димензии, боја на позадина, боја и

дебелина на молив, боја на четка и сл.) и основните операции (за цртање линија, правоаголник и елипса).

Блок (Исказ)	Значење
лизгај 1 секунди до x: 0 y: 0	Поместување на фигурата рамномерно до дефинирана позиција за определено време.
оди до x: 0 y: 0	Поместување на фигурата до дефинирани x и y позиција на Етапата
смени x за 10	Промена на x-позицијата на фигурата за дефинирана вредност
намести x на 0	Поместување на x-позицијата на фигурата за дефинирана вредност
<input type="checkbox"/> x позиција	Прикажување на x-позицијата на фигурата (опсег од 24 до 240)
<input type="checkbox"/> насока	Прикажување на насоката на фигурата (0=нагоре, 90=надесно, -90=налево, 180=надолу)

Активност 1:

Напиши го следниот исказ кој го прикажува на екран графичкиот прозорец:

```
GraphicsWindow .Show ()
Каква е бојата на позадината на графичкиот прозорец спореден со текстуалниот прозорец?
```



Со додавање на следните искази на прозорецот му доделуваме име, боја на позадина и димензии – ширина и висина :

```
GraphicsWindow .Title = "GRAFICKO PROZORCE"
GraphicsWindow .BackgroundColor = „yellow“
GraphicsWindow .Width = 320
GraphicsWindow .Height = 200
GraphicsWindow .Show ()
```

Дали ги препознаваш операциите со кои се доделува име, боја на позадина, висина и ширина? Како ќе го затвориш графичкиот прозорец?

Активност 2: Напиши програма која на екранот црта две линии:

```

GraphicsWindow . Title ="GRAFICKO PROZORCE"
GraphicsWindow .Width =300
GraphicsWindow .Height =300
GraphicsWindow . DrawLine (100, 100, 200,
200)
GraphicsWindow . DrawLine (200, 100, 100,
200)

```



Дали ја препознаваш операцијата која црта линија во графичкиот прозорец?

Воочи: Кога се работи со графичкиот прозорец при цртање се работи со координати на точките од прозорецот, при што точката во горниот лев агол од прозорецот има координати (0,0). Од лево на десно по x оската се зголемува вредноста на x координатата, а од горе надолу по y оската се зголемува вредноста на y координатата.

Воочи: Структурата на операцијата DrawLine е DrawLine (x1,y1,x2,y2) каде x1 и y1 се координати на почетната а x2 и y2 координати на крајната точка на линијата.

Активност 3: Во програмскиот код додаваме искази со кои ја менуваме бојата на линијата и нејзината дебелина:

```

GraphicsWindow . Title ="GRAFICKO PROZORCE"
GraphicsWindow .Width =300
GraphicsWindow .Height =300
GraphicsWindow .PenColor ="red"
GraphicsWindow .PenWidth =(20)
GraphicsWindow . DrawLine (100, 100,
200, 200)

```



```

GraphicsWindow . DrawLine (200, 100, 100, 200)

```

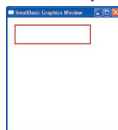
Обиди се да ја промениш програмата со исцртување на линиите во различна боја секоја и со различна дебелина.

Активност 3: Напиши програма која на екранот црта правоаголник:

```

GraphicsWindow .Width =300
GraphicsWindow .Height =300
GraphicsWindow .PenColor ="red"
GraphicsWindow . DrawRectangle (20,
20, 200, 50)

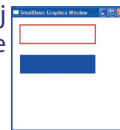
```



Воочи: Операција за цртање правоаголник во графичкиот прозорец е DrawRectangle со структура:

DrawRectangle (x,y,висина,ширина) каде x, y се координати на горната лева точка, а ширина и висина се вредности за ширината и висината на правоаголникот.

Активност 4: Во програмата на крај ќе додадеме искази за исцртување на уште еден правоаголник исполнет со сина боја:
GraphicsWindow . BrushColor ="blue"
GraphicsWindow . FillRectangle (20, 100, 200, 50)



Воочи: Својството за боја на четка во графичкиот прозорец е BrushColor ="боја". Операција за цртање правоаголник исполнет со боја на четката во графичкиот прозорец е FillRectangle со структура: FillRectangle (x,y,висина,ширина) со значење на вредностите во заградите како кај операцијата DrawRectangle .

Обиди се да ја промениш програмата за да ги исцртува и бои правоаголниците со друга боја.

Активност 5: Во програмата замени ги операциите DrawRectangle и FillRectangle со DrawEllipse и FillEllipse и анализирај го излезот! Ке добиеш нацртани две елипси наместо правоаголниците.



Воочи: Операција за цртање елипса во графичкиот прозорец е DrawEllipse, а операција за цртање елипса исполнета со боја на четката во графичкиот прозорец е FillEllipse

Значење на вредностите во заградите е исто како кај претходните операции. Замисли ја елипсата во правоаголник со горна лева координата x,y и најголема хоризонтална дијагонала = ширина, а најголема вертикална дијагонала = висина.

Завршен дел

Активност 6: Обиди се да ја промениш вредностите во заградите во операцијата за цртање и пополнување елипса за да добиеш кружници. Промени ја бојата на четката и дебелината на моливот. Какви се сите дијагонали кај кружницата?

Изработка на програми

Цел на часот:

– ученикот/ученичката да се оспособи да ги користи изучените техники за програмирање при изработка на различни графички прикази.

Програмски јазик Small basic

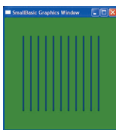
Воведен дел

Дискусија: Досега изучените можности на програмскиот јазик како исказите со услов, циклуси со услов, циклуси со бројач и сл. може да се користат во графичкиот прозорец за исцртување интересни форми – статични или со ефекти на анимација. И во графичкиот прозорец има операции за бришење на прозорецот, паузирање на текот на извршување на програмата и сл.

Главен дел: Пишување на пример програмски кодови за приказ во графичкиот прозорец. Внесување на промени, самостојна изработка на програми по зададен излез и запознавање со нови операции и својства во графичкиот прозорец (случајна боја, случаен број, пауза и бришење).

Активност 1: Напиши го следниот пример програмски код и анализирај го излезот во графичкиот прозорец:

```
GraphicsWindow .Width ="300"  
GraphicsWindow .Height ="300"  
GraphicsWindow .BackgroundColor ="green"  
GraphicsWindow .PenColor ="blue"  
For x=50 To 250 Step 20  
GraphicsWindow .DrawLine (x,50,x,250)  
EndFor
```



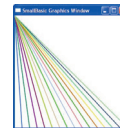
Воочи: За линиите да бидат вертикални и паралелни координатата x се менува а у1,у2 координатите се непроменливи.

Каде треба да го додадеме исказот Program . Delay (100) (паузирање на програмата за 100 msec.) за исцртувањето на линиите да биде постапно? (После исказот во циклусот). Како ќе се постигне ефект на анимација на линиите? (Со додавање на исказ за бришење на екранот GraphicsWindow .Clear () по исказот за пауза во циклусот).

Активност 2:

Направи промени во програмата. Обиди се линиите кои се исцртуваат да бидат хоризонтални и паралелни или да имаат иста почетна точка. Промени им ја бојата, дебелината, позадината. Анимирај го нивното исцртување. Дебелината на секоја следна линија нека биде поголема и слично.

```
GraphicsWindow .Width ="300"  
GraphicsWindow .Height ="300"  
For x=0 To 300 Step 20  
GraphicsWindow . PenColor =Graphics-  
sWindow .GetRandomColor ()  
GraphicsWindow .DrawLine (0,0,x,300)  
Program . Delay (100)  
EndFor
```



Воочи го исказот со кој на моливот му се доделува случајна боја:

PenColor =GraphicsWindow .GetRandomColor (). Наместо конкретна боја на моливот му е придружен исказ со операција на графичкиот прозорец за случајна боја.

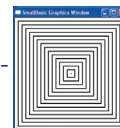
Активност 3: Обиди се да напишеш програмски код за следните прикази. Колку циклуси со бројач треба да користиш? Дали ќе го искористиш циклусот од претходниот пример? Што треба да промениш во вториот циклус, а што ќе остане исто?



Активност 4:

Напиши програма која исцртува концентрични квадрати:

```
GraphicsWindow .Width ="300"  
GraphicsWindow .Height ="300"  
For br=0 To 150 Step 10  
GraphicsWindow .DrawRectangle (150-  
br,150-br,br*2,br*2)  
EndFor
```



Воочи: Прво се исцртува најмалиот внатрешен квадрат (точка) на средината од прозорецот (координати (150,150)). Потоа за секој следен квадрат x и y координатата се намалуваат за иста вредност (чекорот на бројачот во циклусот, а ширината и висината се зголемуваат лево и десно од центарот на прозорецот за вредноста на чекорот – затоа има *2).

Обиди се да го анимираш цртањето на концентричните квадрати.

Активност 5: По пример на претходната програма обиди се да нацрташ концентрични кружници. Само замени ја наредбата DrawRectangle со DrawEllipse. Потоа обиди се да ги анимираш со или без бришење на екранот и со два циклуса со бројач откако ќе се исцрта најголемата кружница да започне повторно исцртување на концентричните кружници кон најмалата.

Активност 6: Напиши го пример програмскиот код кој на различни места на екранот по случаен избор исцртува мали ооени кружници во случајна боја. Во програмата ќе го користиме објектот Math и операцијата GetRandomNumber за доделување на случаен број.

```
GraphicsWindow . Title ="ANIMACIJA SO TOP-CINJA"
```

```
GraphicsWindow .Width ="300"
GraphicsWindow .Height ="300"
GraphicsWindow .BackgroundColor ="black"
For br=1 To 1000
```

```
GraphicsWindow .BrushColor = GraphicsWindow
.GetRandomColor ()
```

```
x= Math .GetRandomNumber (300)
y= Math .GetRandomNumber (300)
GraphicsWindow .FillEllipse (x,y,10,10)
Program . Delay (10)
EndFor
```



Завршен дел

Проверка, дискусија, презентација на успешно изработените програми..

Желка графика

Цел на часот:

- ученикот/ученичката да се запознае со концептот на желка-графика;
- да се оспособи за конструкција на програми во желка-графика.









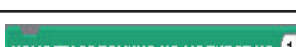

Воведен дел

Дискусија за желка графика како дел од програмските јазици. Желка графиката е компјутерска графика препознатлива по желчето кое се прикажува

во прозорец на екранот и извршува наредби од типот: оди напред, сврти десно, сврти лево и сл. и на тој начин овозможува исцртување интересни форми на екранот. За прв пат е воведена во 1970 год. како дел од едноставниот, но моќен програмски јазик Logo. Желка графиката и денес е дел од некои програмски јазици.

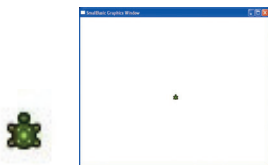
Главен дел: Запознавање со желка графика во програмскиот јазик, неговите својства и операции. Пишување пример програми и анализа на кодот и приказот во графичкиот прозорец.

Активност 1: Програмски јазик Скреч

Блок (Исказ)	Значење
	Чистење на сите траги од молив од Етапата
	Цртање со движење на фигурата
	Запирање на цртање и доколку се движи фигурата
	Поставување боја на моливот
	Промена на бојата на моливот за дефинирана вредност
	Поставување боја на моливот на дефинирана вредност
	Промена на сенката на моливот за дефинирана вредност
	Поставување сенка на моливот на дефинирана вредност
	Промена на големина на моливот за дефинирана вредност
	Поставување на големината на моливот на дефинирана вредност
	Печатење на сликата на фигурата на Етапата

Програмски јазик Small basic

Напиши го исказот: Turtle .Show () кој го прикажува желчето во графичкиот прозорец.



Воочи: Прозорецот има димензии 640x480 и желчето е позиционирано на средина.

Додади ги исказите:

Turtle . Angle =90

Turtle . x=100

Turtle . y=50

Анализирај како се промени положбата и ориентацијата на желката во прозорецот.

Воочи: Желчето има својства: позиција на екранот (x и y) , агол на вртење – ориентација (Angle).

Активност 2: Напиши ја пример програмата во која се исцртува квадрат во прозорецот:

Програмски јазик Скреч	Програмски јазик Small basic
<p>Програмата може да се запише пократко со јамка со бројач:</p> <p>Воочи аголот на завртување е 360/4</p>	<pre>Turtle .Show () Turtle . Move (100) Turtle .TurnLeft () Turtle . Move (100) Turtle .TurnLeft () Turtle . Move (100) Turtle .TurnLeft () Turtle . Move (100)</pre> <p>Програмата може да се запише пократко со јамка со бројач:</p> <pre>Turtle .Show () For br=1 To 4 Turtle . Move (100) Turtle .TurnLeft () EndFor</pre>

Активност 3: Во последниот пример ќе додадеме исказ за промена на брзината на движење на желката и за бојата на моливот:

Програмски јазик Скреч	Програмски јазик Small basic
	<pre>Turtle .Show () Turtle . Speed =8 GraphicsWindow . PenColor = "red" For br=1 To 4 Turtle . Move (100) Turtle .TurnLeft () EndFor</pre>




Направи промена во програмата: За секоја линија од квадратот да биде друга боја користи ја операцијата за случајна боја. Каде ќе го поставиш исказот, во или надвор од циклусот? Објасни!

Активност 4: Обиди се со „управување“ на желката да нацрташ рамностран триаголник со страна =100 и секоја страна да биде во различна боја. Внимавај при изборот на аголот на вртење. Едно можно решение би било:

Програмски јазик Скреч	Програмски јазик Small basic
<p>Пред цртањето позиционирај ја желката на позиција (10,10).</p> <p>Воочи аголот на завртување е 360/3</p>	<p>Пред цртањето позиционирај ја желката на позиција (270, 240).</p> <pre>Turtle .Turn (30) Turtle . Move (100) Turtle . Turn (120) Turtle . Move (100) Turtle . Turn (120) Turtle . Move (100)</pre> <p>Воочи ја операцијата за вртење на желката за агол Turn (агол).</p>

Завршен дел

Активност 5: Обиди се со „управување“ на желката да ја испишеш првата буква од своето име.

Програмски јазик Скреч	Програмски јазик Small basic
<p>Употреби го блокот за промена на големината и сенката на моливот</p>  	<p>Воочи: Желката може да се движи без да црта со операцијата PenUp(). За да продолжи со цртање треба да се внесе операцијата Pen-Down().</p> <p>Пример на програма со желката која ја испишува буквата „H“.</p> <pre>Turtle . Move (100) Turtle . PenUp () Turtle . Move (-50) Turtle . TurnRight () Turtle . PenDown () Turtle . Move (50) Turtle . PenUp () Turtle . TurnLeft () Turtle . Move (50) Turtle . Turn (180) Turtle . PenDown () Turtle . Move (100)</pre> 

Воочи: Чекорот на желката може да биде негативен (желката се враќа назад). Негативен може да биде и аголот на вртење. Кога е негативен, аголот на вртење е обратен од стрелката на часовникот.

Изработка на програми со желка графика

Цел на часот:

– ученикот/ученицата да се оспособи за конструкција на програми во желка-графика.

Воведен дел

Повторување за основните концепти на желка графиката во програмскиот јазик.


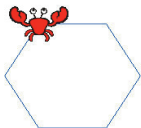
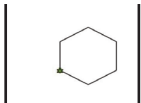
Главен дел: Пример програми за интересни цртања со желката и самостојна изработка на програми со дефиниран излез.

Активност 1: Постапка за цртање правилен n-аголник со желката:

Во програмата се дефинира бројот на страните. Вкупниот периметар на n-аголникот се дели со бројот на страните за да се најде должината на една страна и на крај 360 се дели со бројот на страни за да се пресмета аголот меѓу две соседни страни.

Програмски јазик Скреч	Програмски јазик Small basic
<p>Нацртај правилен 6-то аголник со периметар 500</p>  	<pre>strani=6 perimtar=500 agol=360/strani strana=perimtar/strani Turtle . Speed =9 For br=1 To strani Turtle . Move (strana) Turtle . Turn (agol) EndFor</pre> <p>Воочи ја операцијата за вртење на желката за агол.</p>


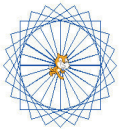
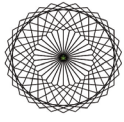
Активност 2: Промени ја програмата да биде интерактивна, односно корисникот да внесува број на страни и периметар на правилниот n-аголник.

Програмски јазик Скреч	Програмски јазик Small basic
<p>расчисти</p>  	<pre>TextWindow .Write („Vnesi go brojot na strani na mnoguagolnikot:“) strani= TextWindow .Read () TextWindow .Write („Vnesi go perimetarot na mnoguagolnikot:“) perimtar= TextWindow . Read () agol=360/strani strana=perimtar/strani Turtle . Speed =9 For br=1 To strani Turtle . Move (strana) Turtle . Turn (agol) EndFor</pre> <p>Воочи дека еден дел од програмскиот код се прикажува во текстуалниот прозореч, а исцртувањата се прават во графичкиот прозореч.</p> 

Активност 3: Ако бројот на страните на n-аголникот е доволно голем желката ќе исцрта круг. Провери.

Завршен дел

Активност 4: Обиди се да ја промениш програмата со додавање на уште еден циклус со бројач за исцртување на серија n-аголници меѓусебно поместени за одреден агол што се постигнува со промена на ориентација на желката за агол кој помножен со бројот на повторувања во надворешниот циклус (бројот на многуаголници) треба да биде 360. На пример: (30- многуаголници * агол од 12 степени = 360 степени)

Програмски јазик Скреч	Програмски јазик Small basic
 	<pre> For k=1 To 30 For br=1 To strani Turtle . Move (strana) Turtle . Turn (agol) EndFor Turtle . Turn (12) EndFor </pre> 

Изработка на програми во желка графика (2 час)

Цел на часот:

- ученикот/ученичката да се оспособи за конструкција на програми во желка графика;
- да го применува стекнатото знаење за изработка на интересни проекти.

Програмски јазик Скреч

Воведен дел:Разговор за желка графика. Бура на идеи за изработка на самостоен проект.

Главен дел: Самостојна изработка на програми во желка графика.

Пример програма: Направи програма со која ќе ја анимираш секоја буква од твоето име. Притоа секоја буква користејќи желка графика ќе исцртува линии во боја.



Како најлесно ќе ја завршиш задачата? Направи ја скриптата само за првата буква, а потоа ископирај ја за сите останати. Единствено нешто што треба да смениш се координатите на секоја буква и бојата на моливот.

Завршен дел: Презентација на направените програми, дискусија.

Наставни содржини од ТЕМА 11: ПОРПРОГРАМИ

Имплементација и примена на потпрограми. Глобална променлива во програмата

Цел на часот:

- ученикот/ученичката да го објаснува поимот потпрограма;
- да разликува потпрограма од главна програма;
- да ги осознае потребата и предностите од користење (повикување) на потпрограми во главната програма;
- да се оспособи за препознавање на проблемски ситуации во кои треба да се примени потпрограма;
- да се оспособи за примена на правилата за именување на потпрограма;
- да се оспособи за правилно користење на променливи од главната програма во потпрограмата.

Програмски јазик Small basic

Воведен дел

Дискусија за проблемска ситуација во која се јаува потреба од повторување на извршување на иста група активности во истиот редослед во различни фази од решавањето на проблемската ситуација.

На пример: во еден ресторан готвачот повеќе пати во денот во подготвува исто јадење по одреден рецепт. Во рецептот може да се промени количината на состојките во зависност од тоа колку луѓе во исти време го нарачале истото јадење. Готвачот не го бара или смислува рецептот секогаш кога има нарачка, туку готви по еднаш испробаниот рецепт секогаш кога е тоа потребно.

При пишувањето на програми многу често се случува истата ситуација.

Главен дел: Запознавање со значењето, потребата, структурата и примената на потпрограма во програмскиот јазик преку пишување и анализа на веќе познати пример програми со додавање на подпрограми (кои работат со променливи од главната програма, се повикуваат во јамки од главната програма и сл.).

Воочи: Една иста група на искази (наредби) се користи во различни делови од програмскиот код во зависност од проблемската ситуација. Оваа група на искази се запишува како посебен дел од главната програма и се нарекува – потпрограма (субрутина). Потпрограма е дел од главната програма кој се состои од низа на искази со кои се извршува одредена активност. Потпрограмите ја намалуваат должината на програмскиот код, го поедноставуваат неговото пишување и разбирање. Потпрограмата има име и може да се повика од каде било во главната програма со одреден исказ. Самата потпрограма обично се пишува под главната програма и има одредена структура со почеток и крај. Името на потпрограмата не е осетливо на мали/големи букви.

Активност 1: Напиши го следниот пример програмски код со 2 потпрограми:

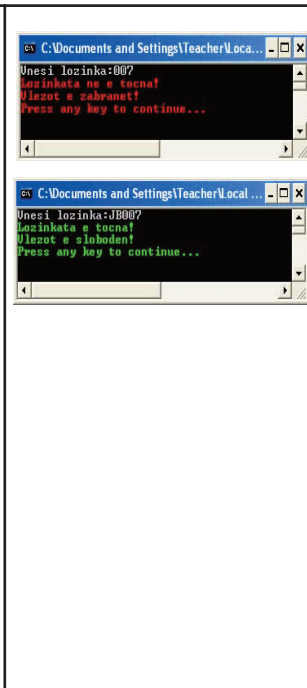
```
TextWindow .Write („Vnesi lozinka:“)
lozinka= TextWindow .Read ()
If lozinka=“JB007” Then
yes()
Else
no()
EndIf
```

Sub no

```
TextWindow . ForegroundColor
=“red”
TextWindow .WriteLine(„Lozinkata ne e točna!“)
TextWindow .WriteLine („Vlezot e zabranet!“)
Sound . PlayBellRing ()
EndSub
```

Sub yes

```
TextWindow . ForegroundColor
=“green”
TextWindow .WriteLine(„Lozinkata e točna!“)
TextWindow .WriteLine („Vlezot e sloboden!“)
Sound . PlayChime ()
EndSub
```



Анализирај го приказот во прозорецот. Дали ја препознаваш програмата од претходните часови? Дали го пишувавме кодот на овој начин? Која е разликата? Дали ги препознаваш подпрограмите со нивните имиња? Дали ги препознаваш исказите со кои се повикуваат потпрограмите во главната програма? Која е структурата која мора да се користи за пишување на потпрограма?

Воочи: Структурата за пишување на потпрограма е следна:

```
Sub име_na_podprogramata
```

```
Искази
```

```
EndSub
```

Потпрограмата се повикува од главната програма со исказот `име_na_podprogramata ()`.

Активност 2: Пишување на пример програма со потпрограма која користи променливи од главната програма. Потпрограмата исцртува вертикални па-

ралелни линии колку што ќе избере корисникот во главната програма.

```
TextWindow.WriteLine („Programa koja crta linii“)  
TextWindow . Write („Kolku linii treba da na-  
crtam?“)  
broj=TextWindow .ReadNumber ()  
linii()
```

```
Sub linii  
For i = 1 To broj  
GraphicsWindow . DrawLine (50+x,50 50+x, 200)  
x=x+20  
EndFor  
EndSub
```

Која променлива од главната програма ја користи потпрограмата? Зошто координатите по x оската мора да се зголемат?

Воочи: Променливите во програмата се глобални, а тоа значи дека ја задржуваат својата вредност и име и во главната програма и во потпрограмата.

Активност 3: Пишување на пример програма која повикува потпрограма во циклус. Програмата е позната и се менува кодот со додавање на потпрограма која исцртува кружница во различна боја на позиција на екранот и онолку пати колку што е дефинирано во главната програма:

```
GraphicsWindow . Title =“ANIMACIJA SO TOP-  
CINJA“
```

```
GraphicsWindow .Width =“300“  
GraphicsWindow .Height =“300“  
GraphicsWindow .BackgroundColor =“black“  
For br=1 To 1000  
x= Math .GetRandomNumber (300)  
y= Math .GetRandomNumber (300)  
kruznicia ()  
EndFor
```

```
Sub kruznicia  
GraphicsWindow .BrushColor = GraphicsWindow  
.GetRandomColor ()  
GraphicsWindow .FillEllipse (x,y,10,10)  
Program . Delay (10)  
EndSub
```

Колку пати се повикува потпрограмата во главната програма? Дали изразот за доделување вредност на координатите x и y може да се внесе во потпрограмата?

Завршен дел: Повторување на новоусвоените поими во врска со потпрограмите и начинот на кој се користат во програмите.

Изработка на програми со потпрограми

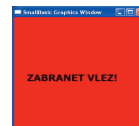
Цел на часот:

ученикот/ученичката да се оспособи за изработка на потпрограми и нивно повикување од главната програма.

Активност. Промени ја програмата за приказот по внесување на лозинката да биде во графичкиот прозорец:

```
TextWindow .Write („Vnesi lozinka:“)  
lozinka= TextWindow .Read ()  
GraphicsWindow .Width =300  
GraphicsWindow . Height = 300  
GraphicsWindow . FontSize = 24  
If lozinka=“JB007“ Then  
yes()  
Else  
no()  
EndIf  
Sub no
```

```
GraphicsWindow .BackgroundColor =“red“  
GraphicsWindow . BrushColor = „black“  
GraphicsWindow . DrawBoundText (30,150, 250, „ZABRA-  
NET VLEZI!“)  
Sound . PlayBellRing ()  
EndSub  
Sub yes  
GraphicsWindow .BackgroundColor =“green“  
GraphicsWindow . BrushColor = „black“  
GraphicsWindow . DrawBoundText (30,150, 250, „SLOBO-  
DEN VLEZI!“)  
Sound . PlayChime ()  
EndSub
```



Воочи: Во графичкиот прозорец може да се пишува текст со одредена големина на фонот и со одредена боја. Операција за пишување на текст во

графичкиот прозорец е DrawBoundText (x,y, ширина, „текст“). Бојата на текстот се дефинира со доделување боја на четката во графичкиот прозорец.

Наставни содржини од ТЕМА 12: НАСТАНИ И ИНТЕРАКТИВНОСТ

Интерактивни програми со настани

Цел на часот:

- ученикот/ученичката да ги објаснува и дефинира поимите објект и настан во објект;
- да ја осознае потребата и предностите од примена на настани за интерактивност во програмата.

Воведен дел





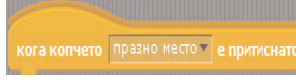
Дискусија за секојдневна ситуација со цел објаснување на настани (случки) како дел од програмските јазици. Ако куќата ја сметаме за објект, бројот на катови, врати, прозорци за особини на објектот, отворање, затворање врати и прозорци за операции во објектот, тогаш што ќе биде звонењето на звончето од влезната врата? Одговорот е настан или случка. На овој настан се одговара со соодветна операција во објектот, на пример отворање на вратата или прашање „Кој е?“. Преку овој настан во куќата се комуницира со надворешната околина.

Главен дел

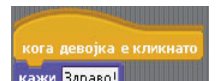

Настаните или случките во програмските јазици претставуваат нов градбен дел кој овозможува интерактивност во програмата преку активности со глумчето или притискање на соодветно копче од тастатурата.

Програмски јазик Скреч

Дали досега работевме со блок кој претставува блок за настан? Воочи дека блок за настан е и „зелено знаме“ со кој претходно се запознавме.

Блокови (искази) за настани	Значење
	Извршување на множеството надолу кога фигурата е кликната.
	Извршување на множеството надолу откако ќе биде примена дефинирана порака.
	Испраќање на порака до сите фигури и продолжување на следен блок без чекање да се активираат множествата.
	Да работат и чекање додека не завршат пред да се продолжи на следен блок. Испраќање на порака до сите фигури, активирање.
	Условен исказ кој овозможува извршување на множеството надолу кога избраното копче од паѓачкото мени е притиснато.

Активност 1.

Програмски јазик Скреч	Програмски јазик Small basic
<p>Напиши програма која ќе овозможи кога ќе се кликне на фигура да се напише порака „Здраво“.</p>	<p>Настаните се во тесна врска со потпрограмите. Потпрограмите се повикуваат кога ќе се случи некој настан и тогаш се извршуваат исказите во нив. Внеси го пример програмскиот код со настан во него кој се опслужува со соодветна потпрограма</p>
 	<pre>GraphicsWindow . MouseDown = klik Sub klik GraphicsWindow . ShowMessage („Ej, pa ti klikna“, „ZDRAVO!“) EndSub Воочи: Операцијата ShowMessage („име_на_прозорецот“, „порака“) прикажува порака во посебен прозорец со доделено име.</pre>

Програмски јазик Скреч	Програмски јазик Small basic
	Воочи: На настанот во графичкиот прозорец му се доделува име на потпрограма која треба да се изврши при активирање на настанот.

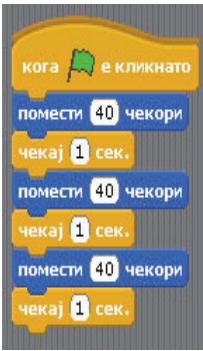

Кој дел е настан? Како го препознаваме? Што му се доделува како вредност?

Активност 2.

Програмски јазик Скреч	Програмски јазик Small basic
При клик на фигурата, таа да се ископира во форма на круг.	При клик со глумчето во прозорецот се исцртува топче на местото каде сме кликнале со глумчето во случајна боја.
 	<pre> GraphicsWindow . MouseDown = klik Sub klik x= GraphicsWindow . MouseX y= GraphicsWindow . MouseY GraphicsWindow . BrushColor = GraphicsWindow . GetRandomColor () GraphicsWindow . FillEllipse (x,y, 10, 10) EndSub Операцијата MouseX и MouseY во графичкиот прозорец ги дава координатите на глумчето во прозорецот. </pre>

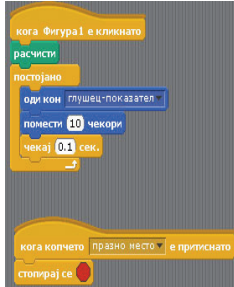

Активност 3. Напиши пример програма со повеќе настани.

Програмски јазик Скреч	Програмски јазик Small basic
Програма со 3 настани – првиот е менување на бојата на топчето кога ќе се активира програмата (зелено знаме), вториот настан е притискање на стрелките од тастатурата при што се менува насоката на топчето, додека третиот настан е расчистување на Етапата кога ќе се притисне тастерот "празно место" од тастатурата.	Настан со клик на глумчето при што се повикува потпрограмата која црта топче, а другиот настан со притискање на било кое копче од тастатурата при што се менува бојата на топчињата кои се цртаат.

 	<pre> GraphicsWindow . MouseDown = klik GraphicsWindow . KeyDown = tipka Sub klik x= GraphicsWindow . MouseX y= GraphicsWindow . MouseY GraphicsWindow . FillEllipse (x,y, 10, 10) EndSub Sub tipka GraphicsWindow . BrushColor = GraphicsWindow . GetRandomColor () EndSub </pre>
---	--

Завршен дел: Пишување програма со настан во која ќе се применува желка графиката.

Активност 4.

Програмски јазик Скреч	Програмски јазик Small basic
Напиши програма во желка графика каде фигурата ќе го следи движењето на покажувачот и притоа ќе ја исцртува патеката на движење. Притискањето на празно место од тастатурата означува крај на програмата.	Напиши програма со настан клик на глумчето по што се активира потпрограма која го движи желчето до позицијата каде е кликнато со глумчето.
 	<pre> Turtle . Show () Turtle.Speed =7 GraphicsWindow.MouseDown = klik Sub klik x = GraphicsWindow.MouseX y = GraphicsWindow.MouseY Turtle . MoveTo (x,y) GraphicsWindow . PenColor =GraphicsWindow . GetRandomColor () EndSub Воочи ја операцијата на желката MoveTo (x,y) со која желката се придвижува до точка со координати x и y. </pre>

Наставни содржини од ТЕМА 13 ЕДНОДИМЕНЗИОНАЛНИ НИЗИ

Дефинирање и декларирање на еднодимензионална низа

Цел на часот:

– ученикот/ученичката да ги објаснува и дефинира поимите структура на податоци и еднодимензионална низа како структура од податоци;

– да ја осознае потребата од користење на структура на податоци – еднодимензионална низа;

– да декларира, дефинира и користи еднодимензионална низа.

Воведен дел: Анализа на пример програма во која се внесуваат повеќе податоци од ист тип кои потоа се користат во програмата

Програмски јазик Скреч	Програмски јазик Small basic
	<pre>TextWindow . Write („Vnesi juzno ovosje br.1:“) ovosje1= TextWindow . Read () TextWindow . Write („Vnesi juzno ovosje br.2:“) ovosje2= TextWindow . Read () TextWindow . Write („Vnesi juzno ovosje br.3:“) ovosje3= TextWindow . Read () TextWindow . Write („Vnesi juzno ovosje br.4:“) ovosje4= TextWindow . Read () TextWindow . Write („Vnesi juzno ovosje br.5:“) ovosje5= TextWindow . Read () TextWindow . Write („Na nasiot pazar se prodavaat slednite juzni ovosja: „) TextWindow .Write (ovosje1+“, „+ovosje2+“,„+ovosje3+“,„+ovosje4+“,„+ovosje5+“,„) TextWindow . WriteLine („“)</pre>

Заклучок: програмскиот код е долг, се повторува истиот исказ за различна вредност на истиот тип на податок.

Решение: наместо да се користи нова променлива за секој од податоците, податоците може да се сместат во иста променлива која ќе додава различна ознака за секој нов податок.

Главен дел: Запознавање со поимот низа (листа, поле), неговото значење и примена при пишување програма во програмскиот јазик.

Променливата која може да има повеќе од една вредност кои се различно означени се вика еднодимензионална низа или еднодимензионално поле. Означките на податоците може да бидат бројни или текстуални и се викаат индекси. Податоците во низата обично се викаат и членови на низата. Вредностите на членовите во низата може да бидат броеви или текстови.

Активност 1: Внеси ја пример програмата запишана со помош на низа и анализирај го кодот и излезот. Можно решение:

Програмски јазик Скреч	Програмски јазик Small basic
	<pre>For i=1 To 5 TextWindow . Write („Vnesi juzno ovosje br.“+i+“: „) ovosje[i] = TextWindow . Read () EndFor TextWindow . Write („Na nasiot pazar se prodavaat slednite juzni ovosja: „) For i = 1 To 5 TextWindow . Write (ovosje[i]) If i<5 Then TextWindow . Write („ „) EndIf EndFor TextWindow . WriteLine („“)</pre>

Воочи: i е променлива, додека низа е листа.

Кое е името на низата? Како се означени членовите на низата? Колку членови има низата во програмата? Какви се вредностите на членовите: текстуални или бројни?

Активност 2. Напиши програма која внесува податоци за име, презиме и година на раѓање на 5 ученици од едно училиште. Податоците се внесуваат во три низи и, по потреба, се користат во програмата:

Програмски јазик Скреч	Програмски јазик Small basic
	<pre> For i=1 To 5 TextWindow .Write („Vnesi ime na ucenikot br.“+i+“ :„) ime[i]= TextWindow .Read () TextWindow .Write („Vnesi pre- zime na ucenikot:“) prezime[i]= TextWindow .Read () TextWindow .Write („Vnesi oценка po informatika na uceni- kot.“) oценка[i]= TextWindow .Read- Number () EndFor TextWindow .Clear () For i= 1 To 5 TextWindow .Write („Vnesi go redniot broj od 1 do 5 za ucenikot za koj sakas podatoci:„) indeks= TextWindow .ReadNum- ber () TextWindow .WriteLine („Uceni- kot „+ime[indeks]+“ „+prezime[indeks]+“ ima oценка „+oценка[indeks]+“ po informa- tika“) EndFor </pre> <p>Зошто е потребен исказот за бришење на екранот по внесување на податоците за членовите од низата?</p>

Активност 3.
Програмски јазик Скреч

Подобри ја претходната програма со внесување на услови (пр., годината на раѓање да припаѓа на интервалот 1994 до 1998).

Програмски јазик Small basic

Напиши програма со две низи, која во една низа сместува имиња на држави, а во друга имиња на главни градови, го брише екранот и по случаен избор поставува прашање за главен град на некоја држава:

```

For i=1 To 5
  TextWindow .Write („vnesi drzava:“)
  drzava[i]= TextWindow .Read ()
  TextWindow .Write („Vnesi glaven grad:“)
  grad[i]= TextWindow .Read ()
EndFor
TextWindow .Clear ()
For i=1 To 5
  indeks= Math .GetRandomNumber (5)
  „+drzava[indeks]+“ ? „)
  odgovor= TextWindow .Read ()
  If odgovor = grad[indeks] Then
    TextWindow .WriteLine („Odgovorot e tocen!“)
  Else
    TextWindow .WriteLine („Odgovorot ne e tocen!“)
  EndIf
EndFor

```

Завршен дел: Дискусија по задачите. Програмата се состои од два дела: внесување на вредности за членовите од низата и користење на вредностите.

Наставни содржини од ТЕМА 14
ПРИМЕРИ ЗА ПОСЛОЖЕНИ АЛГОРИТМИ И ПРОГРАМИ

Програми за познати и посложени проблемски ситуации

Програмски јазик Скреч

1. Напиши програма која ги испишува првите 16 члена од Фибоначиевата низа.

str 1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987



Програмски јазик Small basic

1. Напиши програма која прикажува анимација во вид на топче кое се одбива од границите на прозорецот и при секое одбивање ја менува бојата.

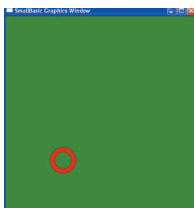
```
GraphicsWindow .Width =450  
GraphicsWindow .Height =450  
GraphicsWindow .PenWidth =10  
For m=1 To 5
```

```
For x=0 To 200 Step 10  
GraphicsWindow . PenColor ="red"  
GraphicsWindow . BackgroundColor ="green"  
GraphicsWindow .DrawEllipse (0+x,200+x,50,50)  
Program .Delay (50)  
GraphicsWindow . Clear ()  
EndFor
```

```
For x=0 To 200 Step 10  
GraphicsWindow . PenColor ="black"  
GraphicsWindow . BackgroundColor ="green"  
GraphicsWindow .DrawEllipse (200+x,400-x,50,50)  
Program .Delay (50)  
GraphicsWindow . Clear ()  
EndFor
```

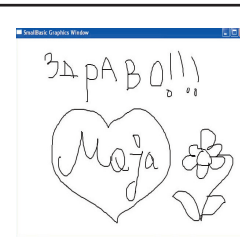
```
for x=0 To 200 Step 10  
GraphicsWindow . PenColor ="blue"  
GraphicsWindow . BackgroundColor ="green"  
GraphicsWindow .DrawEllipse (400-x,200-x,50,50)  
Program .Delay (50)  
GraphicsWindow . Clear ()  
EndFor
```

```
For x=0 To 200 Step 10  
GraphicsWindow . PenColor ="yellow"  
GraphicsWindow . BackgroundColor ="green"  
GraphicsWindow .DrawEllipse (200-x,x,50,50)  
Program .Delay (50)  
GraphicsWindow . Clear ()  
EndFor  
EndFor
```



3. Напиши пример програма "Paint" за настан во графичкиот прозорец со движење на глумчето, но ако е притиснат левиот клик:

```
GraphicsWindow . MouseMove =  
crtaj  
GraphicsWindow . MouseDown =  
pocni  
  
Sub pocni  
pocetnox= GraphicsWindow . MouseX  
pocetnoy= GraphicsWindow . MouseY  
EndSub  
Sub crtaj  
x= GraphicsWindow . MouseX  
y= GraphicsWindow . MouseY  
If (Mouse . IsLeftButtonDown ) Then  
GraphicsWindow . DrawLine (pocetnox,  
pocetnoy, x,y)  
EndIf  
pocetnox=x  
pocetnoy=y  
EndSub
```



Воочи го новиот објект кој се користи во програмата Mouse и неговото својство IsLeftButtonDown.