

1.Սահմանել փոփոխական:
Փոփոկանալը հիշողության մեջ
անվանված տիրույթ է

2.Սահմանել տիպ:
Արժեքների բազմություն է, որի վրա
սահմանված են տարրական
գործողություններ:

3.Սահմանել ֆունկցիա:
Ֆունկցիան կախվածություն է ինչ-որ
բանից:

4.Կոմպիլյատորների տեսակներ:
GCC, Clang , MSVC

5.OOP

Օբյեկտ կողնորոշված
ծրագրավորումը ծրագրավորման
պարադիգմա է, հիմնված օբյեկտների
հայեցակարգի վրա, պարունկում է

տվյալներ և ֆունկցիոնալություն:

1)Աբստրակցիա:

Իրականության պրոեկցիան է:

2)Ինկապսուլյացիա:

Օբյեկտի ինֆորմացիայի

կենտրոնացումը մի կետի շուրջ:

3)Ժառանգում:

Կլասսի հնարավորությունը

փոխանցել իր ժառանգ կլասսին, զերծ

մնալով կոդի կրկնությունից:

4)Պոլիմորֆիզմ:

Կոդի մասի ներկայացումը տարբեր

ինտերֆեյսներով:

6.Ինչ է դասը:

Դասը օբյեկտի նախագիծն է:

7.Ինչ է օբյեկտը:

Հիշողության բլոկ է,որը պարունակում է

արժեքավորված տվյալների դաշտեր:

8.Դասի կառուցվածքը:
Կազմված է տվյալներից և
ֆունկցիաներից:

9.Ցուցիչի օգտագործումը դասի
դաշտերի վրա:
Ցուցիչի միջոցով կարող են հղում
կատարել տվյալ օբյեկտի
անդամներին:

10.Կոնստրուկտորներ և օբյեկտների
սկզբնավորում:

Դասի անունը կրող ֆունկցիա, որը
չունի վերադարձվող արժեք,
սկզբնավորում է դասի օբյեկտի
դաշտերը ստեղծման պահին :

1.Default constructor

2.constructor with parameters

3.copy constructor

11.Լռելյայն կոնստրուկտոր:

Եթե չկա սահմանված կոնստրուկտոր, կոմպիլյատորն կազմում է լռելյայն կոնստրուկտոր, որը ոչ մի պարամետր չի ընդունում և դաշտերին տալիս է default արժեքներ:

12.constructor delegation

Կոդի քանակը նվազեցնելու համար, կարող ենք կանչել մի կոնստրուկտորից մյուսը:

13.Ինիցիալիզացիա ցանկ:

Սկզբնավորիչների թվարկումներ յուրաքանչյուր փոփոխականի և հաստատունի համար, որոնք առանձնացված են 2 կետով կոնստրուկտորի պարամետրիի ցանկից հետո:

14.Մուտքի սպեցիֆիկատորներ:

Սրանք օգտագործվում են դասի

անդամների հասանելիությունը
տարբերակելու համար:

Public- անդամի հասանելիություն կողի
ցանկացած մասում:Թերությունը , որ
կարող են արժեքավորել ոչ ճիշտ
արժեքներով, խաթարելով ծրագրի
տրամաբանական ալգորիթմը:

Private- անդամի հասանելիությունը
միայն տվյալ դասում:Թերությունը
անդամները օգտագործելու
սահմանափակում:

Protected- անդամի հասանելիությունը
տվյալ դասում և իր ժառանգներում:

15.Getter,Setter:

Եթե ցանկանում ենք , որ private
անդամներին հասանելիություն տրվի,
ապա կարող ենք սահմանել get և set

մեթոդները:

Setter-ների միջոցով կարող ենք private արժեքների հետ ինչ-որ փոփոխություն իրականացնել, իսկ getter-ի միջոցով այդ արժեքը ստանալ:

16.declaration & implementation

Կարող են ֆունկցիաները սահմանել կլասսում ,բայց իրականացումը կատարել կլասից դուրս:

17.copy constructor

Պատճենող կոնստրուկտորը պատճենում է օբյեկտի դաշտերի արժեքները նոր օբյեկտի դաշտերում: Թերությունն է , երբ դաշտը ներկայացնում է ցուցիչ,ապա հասցեն պատճենվում է:Արդյունքնում 2 օբյեկտ ցույց են տալիս հիշողության նույն հասցեի վրա:

Քոփի կոնստրուկտորի պարամետրը

պետք է տալ հղումով,քանի որ որպես
արժեք փոխանցելու պարագայում
կոմպիլյատորը պետք է ստեղծի
օբյեկտի պատճեն, իսկ դրա համար
պետք է կանչվի կրկին քոփի
կոնստրուկտորը,որը կհանգեցնի
անվերջ ռեկուրսիայի:
Քոփի կոնստրուկտորը կարող ենք
ջնջել delete օպերատորով:

18.Խորը պատճեն,մակերեսային
պատճեն:

Մակերեսայինի ժամանակ
պատճենվելուց հետո, ռեֆերենս
դաշտերի հասցեները չեն
պատճենվում , հղումները կատարվում
են նույն հասցեների վրա, իսկ
Խորը պատճենի ժամանակ նաև
պատճենվում են ռեֆերենսները,
հղումները կկատարվեն տարբեր
հասցեների վրա , այսինքն մեկի

փոփոխությունից մյուսի արժեքը չի փոխվի:

20.Հաստատուն օբյեկտ:

Երբ օբյեկտը հաստատուն է , ապա միայն կարող ենք օգտագործել դրանք, փոփոխել չենք կարող:

21.հաստատուն օբյեկտի մեթոդի կանչ:

Հաստատուն օբյեկտից չենք կարող ոչ կոնստանտ մեթոդներ կանչել ,քանի որ հնարավոր է ,որ մեթոդների ներսում փոփոխվեն օբյեկտի դաշտերի արժեքները:

Կարող ենք կանչել միայն կոնստանտ մեթոդներ,որոնց ներսում դաշտերի արժեքների փոփոխություն չի արվում:

Այս մեթոդները կարող ենք կանչել նաև ոչ կոնստանտ օբյեկտների վրա:

Հաստատուն մեթոդների մեջ միայն կարող ենք կանչել հաստատուն

մեթոդներ և անդամներ:

22. հաստատունի վերադարձ

Կոնստ ֆունկցիայում վերդարձվող արժեքը հղման պարագայում պետք է լինի կոնստանտ, իսկ ցուցիչի դեպքում պետք է ցույց տա դեպի հաստատուն:

23. mutable

Երբ հաստատուն օբյեկտի դաշտերը կրում են mutable keyword-ը, ապա դրանք կարող են փոփոխվել:

24. this keyword

Սա ցուցիչ է, որը ցույց է տալիս ընթացիկ օբյեկտի վրա: Կարող են հասանելություն ունենալ դասի դաշտերի վրա:

25. Friend ֆունկցիաներ և դասեր

Այս ֆունկցիաները դասի անդամ չեն

,բայց հասանելիություն ունեն դասի
private անդամներին:

Կարող են սահմանվել մեկ այլ դասում:

26.Ստատիկ դաշտեր

Ստատիկ անդամները վերաբերում են
ոչ թե դասի օբյեկտներին այլ հենց
դասին:Ստատիկ դաշտերը պահում են
ամբողջ դասի վիճակը:Սրանք
գոյություն ունեն , եթե անգամ օբյեկտ
ստեղծված չի:

Ստատիկ անդամները հասանելի են և
ստատիկ,և ոչ ստատիկ մեթոդներում,
իսկ ստատիկ մեթոդներում հասանելի
են միայն ստատիկ անդամները:

28.զանգված

29.LinkedList

30.big O

31.խնդիրներ

