

2.1) Напечатайте типы данных

int (4б)	long (8б)	char (2б)
byte (1б)	double (8б)	boolean (1б)
short (2б)	float (4б)	

2.2) Напишите цикл for, который проходит по

массиву в обратном порядке - от последнего  
элемента к первому

```
int [] arr = { 2, 5, 8, 9 };
```

```
for ( int i = arr.length - 1, i >= 0, i -- ) {  
    cout << arr[i];  
}
```

### 3.1) Чим маєте "класи"

Клас (Java class) - структура, що містить даних та методів, які виконують определені функції (methods). Класи можуть бути наслідувані іншими класами.

### 3.2) Напишіть цикл while, який виведе всі елементи масиву

```
int [] arr = { 2,5,8,9 }  
int index = arr.length-1;  
while ( index >= 0 ) {  
    cout ( arr [index] );  
    index = index - 1; // index -- ;  
}
```

4.1) Тип enum (написанное) - что это такое?

Списочный тип данных, который содержит множество элементов определенных значений.

public enum Places { KYIV, KHARKIV }

А a = new (A()); a.setPlace (Places.KYIV);

4.2) Создание класса с 2 полями - одиноковым и множественным и присвоение им значений для конкретных полей

@Geter  
@Setter

public class Student {

private String name;

private int age;

Student s = new Student();

s.setName ("Den");

s.setAge (19);

5.1) Задача наследия классов и методы классов ?

Классы есть "материнки" где конструирование производится.

Методы класса - структура, выполняющие определенную задачу, при необходимости возвращают результат. Как правило, минимизируют самоподдержку обертка.

5.2) Напишите фрагмент, который с помощью цикла проходит по массиву студентов и выводит на экран имя каждого студента

```
Student [] arr = { new Student ("Den", 20),  
new Student ("Mary", 15),  
new Student ("Nick", 20) };
```

```
for ( Student s : arr ) {
```

```
    sout (s);
```

```
}
```

6.1) Зарел мисалын жолчыларын?

Конструктордың негизгіліктерінде (предикторларда) | зерттеу  
неше орнеки нұсқаудардың орнеки нұсқаударынан, Орнеки негизгіліктерінде  
конструктордың сокреттілдік конст-р нө уйнанылған, еро  
мәдени орнеки нұсқаудардың нө мәдени орнеки нұсқаударынан

6.2) Нұсқаудардың орнеки нұсқаударынан және конструктордан

public class Animal {

private int age;

private double weight;

public Animal () { }

public Animal (int age, double weight) {

this.age = age;

this.weight = weight;

}

.....

Animal a = new Animal();

Animal a1 = new Animal (10, 8.4);

7.1) = 5.1

7.2) = 5.2

7.3) В глине различают механическим и не  
химическим методом

Способные методы не характеризуют состоянием  
материала, например, его физическими свойствами. Ра-  
ботают со структурой почвы.

7.4) Когда метод можно обозначить химическим, а  
когда не химическим?

Метод обозначается химическим, когда результаты его  
работы не будут зависеть от природы, т.е.,  
принадлежности классу. Нечематических, в отличие,  
от спиртосных, методы определяют состояние  
вещества.

8.1) Наиболее важные синхронизированные методы и конструкторы с их описанием. Приведите пример использования синхронизированных методов

написать 6.2.

8.2) Наследование методов базового класса Object

wait(), clone(), equals(),  
notify(), finalize(), hashCode(),  
notifyAll(), toString(),  
getClass()

9.1) Описать 1 одинакий класс в его наследника

```
public class Person {  
    private String name;  
    private int age;  
}
```

```
public class Student  
extends Person {  
    private int course;  
}
```

9.2) Описать наследника от основного класса

	составные	абстрактные методы
класс (обычный)	✓	✗
интерфейс	✗	✓

Класс имеет составные и определенные методы,  
а интерфейс от наследника, который  
определяет в основном абстр. методы  
(несколько default methods и final fields)

10.1) Многопоточное программирование с помощью многозадачного модуля Win32

abstract  
multi class AR  
private A(1){  
}

- abstract. класс

- прив. конст.

10.2) Реализация многопоточности с помощью языка Java

wait()  
notify()  
notifyAll()  
finalize()  
clone()  
getClass()  
equals()

hashCode()

toString()

II.1) Наименование определения ошибка в методе и как исправить ошибку

```
public interface IConverter {  
    public class A implements  
        IConverter {  
        void convert();  
    }  
}
```

```
@override  
public void convert{}  
}
```

II.2) Чему может быть равен адрес переменной?

- Константам, в выражениях которых можно даступить непосредственно, например, в конструкторах.

class scope

method scope

block scope

package scope

12.1) Особенности обертывания класса в наборы  
написаны обобщенные методы в контексте -  
изучение это в коллекциях.

public class A<T> {

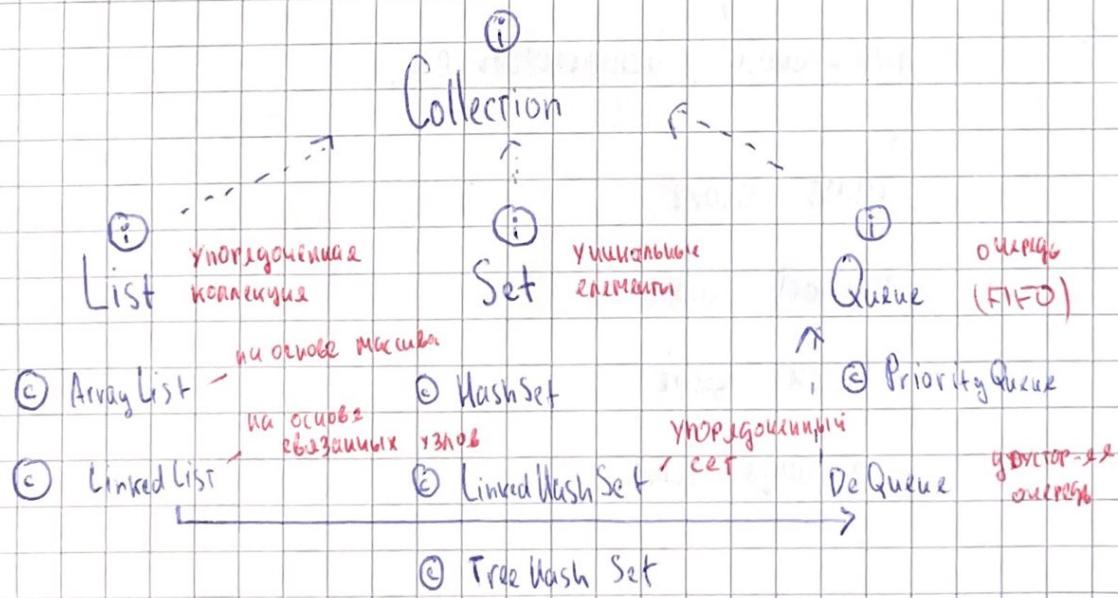
private T t;

public A<T>(T t) {

this.t = t;

}

12.2) Какие виды и типы коллекций бывают?  
и, в чем особенности?



+ Структуры отображения Map (HashMap, LinkedHashMap, TreeMap),  
ключ - значение

13.1) Напишите метод `void update()` с 1 параметром

// Functional Interface

```
public interface SmartHomeConnection {
```

```
    connect();
```

```
}
```

13.2) Запишите метод `-` класса / метода

- в классе, когда класс хранит более типа  $T$ ,

где  $T$  - обобщенный тип, который становится

конкретным при создании экземпляра. Например,

если

в классе `Bag<T>` можно пополнить `item`

множество `String`, в другом экземпляре типа `Integer`

и т.д.

14.1) Составьте программу switch списка и использовать в  
этих методах сравнения.

```
List <String> strings = new ArrayList <> ;  
strings.add ("s1");  
strings.add ("s2");  
strings.add ("s3");
```

14.2) Зарече несколько переопределение equals(), hashCode()  
в наследниках классов и отгадайте?

- Переопределение новых методов позволяет отреагировать  
уникальность объекта, что важно в контекстах и  
картах. (использование хранящихся объектов)
- Для Set для хранения (прививки) хранят объекты
- Для Map для хранения пары отображение, состоящая  
из ключа, сохранности данных.

15.1) Cogganime lezgimup omdorcasche qubix eched  
na ciprosim u nuswaxim b who such  
zurxim.

- Map < Integer, String > map1 = new HashMap<>;  
map1. put (10, "S1");  
map1. put (1, "S10");

15.2) B illi puzgulga menaqy, ArrayList, LinkedList.

ArrayList

oquban na maccube

Basmak : nayaro ( $O_1$ ),

sepeyma ( $O_n$ , nraue),

key (O<sub>1</sub>)

LinkedList

oquban na qubvezancho  
chukke, na yulax

Basmak : nayaro,

menaqy = O<sub>1</sub>, sepeyma

O<sub>n</sub>

Zauqalmam nayaralim yulax  
namiga

yulax zauqalmot  
panqashim yulax  
namiga

16.1) Определение 1 функциональной зависимости в классе  
своих методов

```
public interface IFunc { } | IFunc iFunc = (a) → a/10;  
void m1(double) | ifunc.m1(80.0f);  
}  
|
```

16.2) Реализации в наследуемом классе, которые  
нужны для выполнения

throws - вызывает метод throws, укр-р

throw - спускает укр-л

try - делает метод throws & укр-л

catch - операторы укр, делает throws try

finally - гарантирует делает throws

try-catch-finally, делает throws

б макро определение б не заменяется

б( горо буллюстрирует на упрощение

или нет

17.1) Создайте собственный класс, реализующий методы Comparator <String>.

Comparator <String> c1 = new Comparator <String> {

@Override  
public int compare (String s1, String s2) {

    return Integer.compare (s1.length(), s2.length());

17.2) Каково значение поля - библиотека

IO (Input, Output) назначается для

работы с консолью, файлами, устройствами

и записью информации в базы

мультимедиа, экран, трансляции, работы с

данными

18.1) Створіть об'єкт класа FileOutputStream, змініть  
6 рядок 1 функції в зупинку

```
try { FileOutputStream fos = new FileOutputStream  
(filename + ".txt") {  
    fos.write(1);  
}
```

19.2) Созгайме мэйнчай Map<long, String>.

Наклониме мэйн, засенни.

```
Map<long, String> map = new HashMap<>();
```

```
map.put(1, "Apple");
```

```
map.put(2, "Pineapple");
```

```
map.put(3, "Grape");
```

19.3) Чима маюе JAR - file

Java archive — спорудам архива, иштепелүүлүү б

Java код чындоону, наңысчыпашину нафса

кодчооб, субфайлдар. Реквизити .jar.

20.2) Узнать Set<long> отличие от List<long>

Set - множество, которое хранит некоррелированно  
изменяющиеся уникальные объекты (для определения  
уникальности используется equals(), hashCode()).

Множество не упорядочено, оно может включать  
различные элементы, различающиеся только  
по хэшам.

List - упорядоченное множество, хранит ноды  
(unique, not unique), содержит объекты  
которые имеют no unicity. Реализации  
(основные) : на основе массива, использующего  
массив узлов (arraylist, linkedlist)

21.1) Чима такое Gradle, зачем он нужен.

gradle - это инструмент для работы с проектами, который использует maven repository, поддерживает манипуляции с зависимостями (gradle, gradle), и имеет собственный язык на groovy.

21.2) Напишите Comparator с наименованием reverseComparator<String> и который будет возвращать

Comparator<String> reverseComparator =

```
new Comparator<String>() {
```

② override

```
public int compare (String o1, String o2) {
```

```
    return o2.compareTo(o1)
```

```
}
```

22.1) Чимо малое алигаторама?

Былое нее задачи ногодом, отличного от последовательного. Выполнение "параллельно", не в порядке очереди

22.2) Одноклассник обнаружил вину в грабеже  
мужчину ставшего жертвой, который находился в  
качестве.

```
public class Bag <T> {  
    private T item;  
    Bag (T t) {  
        this.item = t;  
    }  
}
```

23.1) Runnable - згенуу илрүү?

Илрүүнчүүкээс кога б одолдук. Runnable илрүүн дээр  
бийнхэн с нийтийн Executor ямар тулгоо илүү нэгжийн б  
ийн thread. Далгын бийнхэн б одолдук; газаруулж  
ишигдэж. Нэг бийнхэн нэгжийн pr3mises, Future<?>

23.2) Нийтгүүгээ нийтийг онцлогчилж нориносаа гэе  
жүүдээ чиглэвээ

```
try (FileInputStream fis = new FileInputStream(  
    (filepath)) {  
    int data;  
    while (data = fis.read() != -1) {  
        sout(data);  
    }  
}
```

24.1) Who makes Callable ?

но же, это и Runnable (23.1), но老百姓ам  
нельзя выдавать биномиум Future<T>

24.2) Kahan ulmug Future həmşəim qıdınanın  
şəhərindən?.

future.get()

24.3) Mühle paarzähnig unregelmäßig Rundblatt  
class

```
public Runnable CustomTask implements Runnable {
```

@ override

public void run() {

sout ("smth").

3

25.2)

Nyublaume koponuunin wñññep omñññu

Future

try {

long r = future.get(1, sec)

sout(r);

} catch (InterruptedException e) {

future.cancel(true);

sout("task was interrupted");

} catch (TimeoutException t) {

future.cancel(true);

}

If(Thread.currentThread()  
isInterrupted()) {

break;

}

26.1) Maputuline nognomini imputul' kumulule zayam

oawuxhawos e onkayaluk eel neykhama

Executor Service ex = Executors.newCachedThreadPool(),

Future<long> res1 = ex.submit(task1);

Future<long> res2 = ex.submit(task2);

Future<long> res3 = ex.submit(task3);

try {

long result1 = res1.get();

long result2 = res2.get();

} catch (Exception e) {

26.2) Maputuline nognomni halacca thread

Thread.yield

Thread.sleep

Thread.start()

Thread.join

Thread.currentThread().stop()

Thread.currentThread().isAlive()

Thread.currentThread().isInterrupted()

27.1) Káville módosított részleges atomikus környezet

- Végsőkig a Race Condition gyakorlati formájában tűnik itt, amikor több mint egy személyszámú felhasználó

27.2) Káville módosított részleges k.c.h. Volatile

- módosított részleges (számos más részleghez is hozzájárhat, de minden részleghez csak egyetlen módosítás van lehetősége).
- De ha emiatt parancs címére több mint egy személyszámú felhasználó

27.3) Káville módosított részleges synchronized

- Race condition
- csatlakoztatott binomialek

28.1) Muutoksiin seuraavien koodit ovat luotettavat

```
synchronized (lock1) {
```

```
    synchronized (lock2) {
```

```
}
```

```
synchronized (lock2) {
```

```
    synchronized (lock1) {
```

```
}
```

28.2) Neuvontatehtävät ovat muodostamassa Future

Future.get()

Future.get (long, TimeUnit)

Future.cancel (Boolean)

future.isDone (Boolean)

future.isCancelled (Boolean)

25.2) Waéne naypulsa mercau MySQL, SQLite  
c mancau spelle JDBC

- inilauan

- omilirwahme monbuo nyt6 connection

cnypowu. SQLite ne qpredet napow u  
norun bhangenya dg.

31.1) Kourue sysitembejom bayduunur cozgaahe statement  
qal elhammashat SQL zapucca o amayshshuu?

- - Hanpe myro ( s.execute ("DROP TABLE" + name) )
- String Format ( s.execute (StringFormat ( "DROP TABLE", "%s" , name ) ) ) ,

31.2) Muilegumne nikkilip wibedarna meek uchshshuu  
harjous mura qayillu try-catch dildoruu

```
try {  
    Future<Void> future = ...  
    future.get(10, TimeUnit.SECONDS);  
}  
catch (TimeoutException e) {  
    System.out.println(e);  
}  
catch (InterruptedException e) {  
    e.printStackTrace();  
}
```

32.1) Інші варіанти @ After, @ Before ніж у

JUnit

виконувати код (ноготкова, дополнительна)

@ after - як місця (напр. відкритий conn.)

@ before - місце тесту (напр. закрита реєстрація)