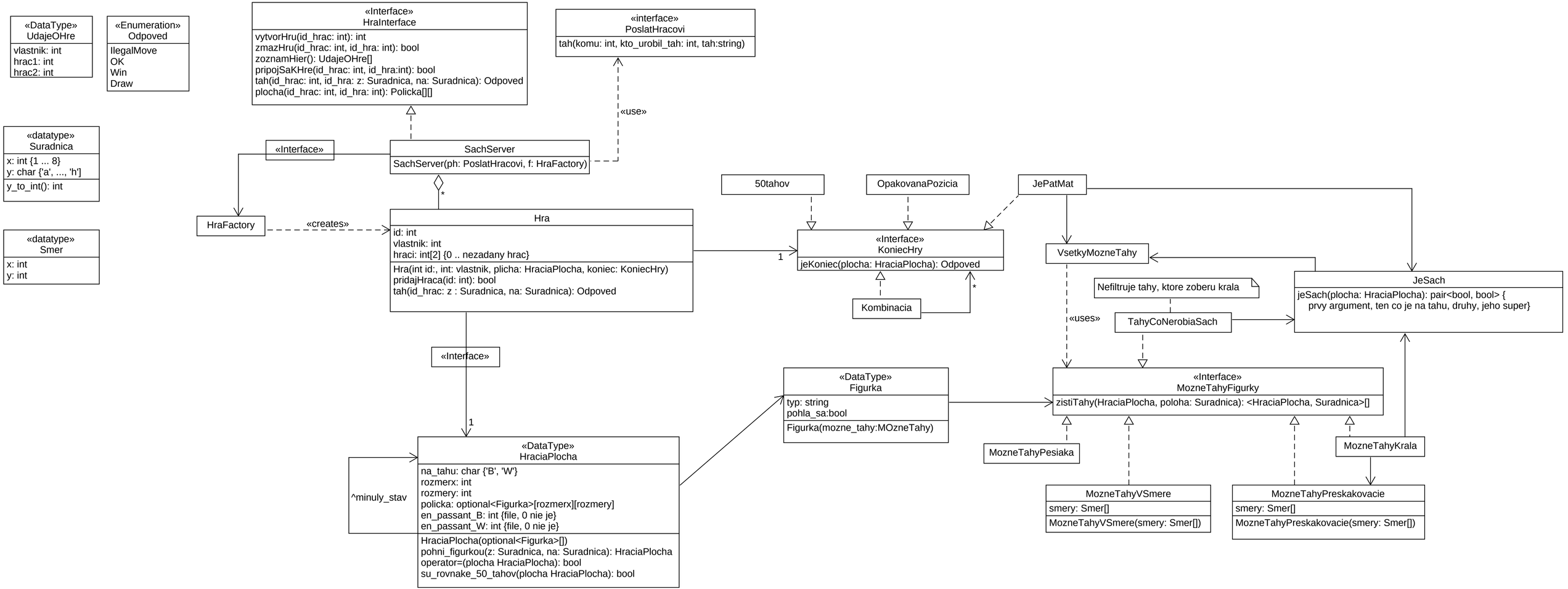
«DataType»
UdajeOHre
vlastnik: int
hrac1: int
hrac2: int
«Enumeration»
Odpoved
IlegalMove
OK
Win
Draw
«Interface»
HraInterface
vytvorHru(id_hrac: int): int
zmazHru(id_hrac: int, id_hra: int): bool
zoznamHier(): UdajeOHre[]
pripojSaKHre(id_hrac: int, id_hra:int): bool
tah(id_hrac: int, id_hra: z: Suradnica, na: Suradnica): Odpoved
plocha(id_hrac: int, id_hra: int): Policka[][]
«interface»
PoslatHracovi
tah(komu: int, kto_urobil_tah: int, tah:string)
«use»
«datatype»
Suradnica
x: int {1 ... 8}
y: char {'a', ..., 'h'}
y_to_int(): int
«Interface»
SachServer
SachServer(ph: PoslatHracovi, f: HraFactory)
*
HraFactory
«creates»
Hra
id: int
vlastnik: int
hraci: int[2] {0 .. nezadany hrac}
Hra(int id:, int: vlastnik, plicha: HraciaPlocha, koniec: KoniecHry)
pridajHraca(id: int): bool
tah(id_hrac: z : Suradnica, na: Suradnica): Odpoved
«datatype»
Smer
x: int
y: int
50tahov
OpakovanaPozicia
JePatMat
«Interface»
KoniecHry
jeKoniec(plocha: HraciaPlocha): Odpoved
1
*
Kombinacia
VsetkyMozneTahy
Nefiltruje tahy, ktore zoberu krala
JeSach
jeSach(plocha: HraciaPlocha): pair<bool, bool> {
prvy argument, ten co je na tahu, druhy, jeho super}
«uses»
TahyCoNerobiaSach
«Interface»
1
«DataType»
Figurka
typ: string
pohla_sa:bool
Figurka(mozne_tahy:MOzneTahy)
«Interface»
MozneTahyFigurky
zistiTahy(HraciaPlocha, poloha: Suradnica): <HraciaPlocha, Suradnica>[]
^minuly_stav
«DataType»
HraciaPlocha
na_tahu: char {'B', 'W'}
rozmerx: int
rozmery: int
policka: optional<Figurka>[rozmerx][rozmery]
en_passant_B: int {file, 0 nie je}
en_passant_W: int {file, 0 nie je}
HraciaPlocha(optional<Figurka>[])
pohni_figurkou(z: Suradnica, na: Suradnica): HraciaPlocha
operator=(plocha HraciaPlocha): bool
su_rovnake_50_tahov(plocha HraciaPlocha): bool
MozneTahyPesiaka
MozneTahyKrala
MozneTahyVSmere
smery: Smer[]
MozneTahyVSmere(smery: Smer[])
MozneTahyPreskakovacie
smery: Smer[]
MozneTahyPreskakovacie(smery: Smer[])