

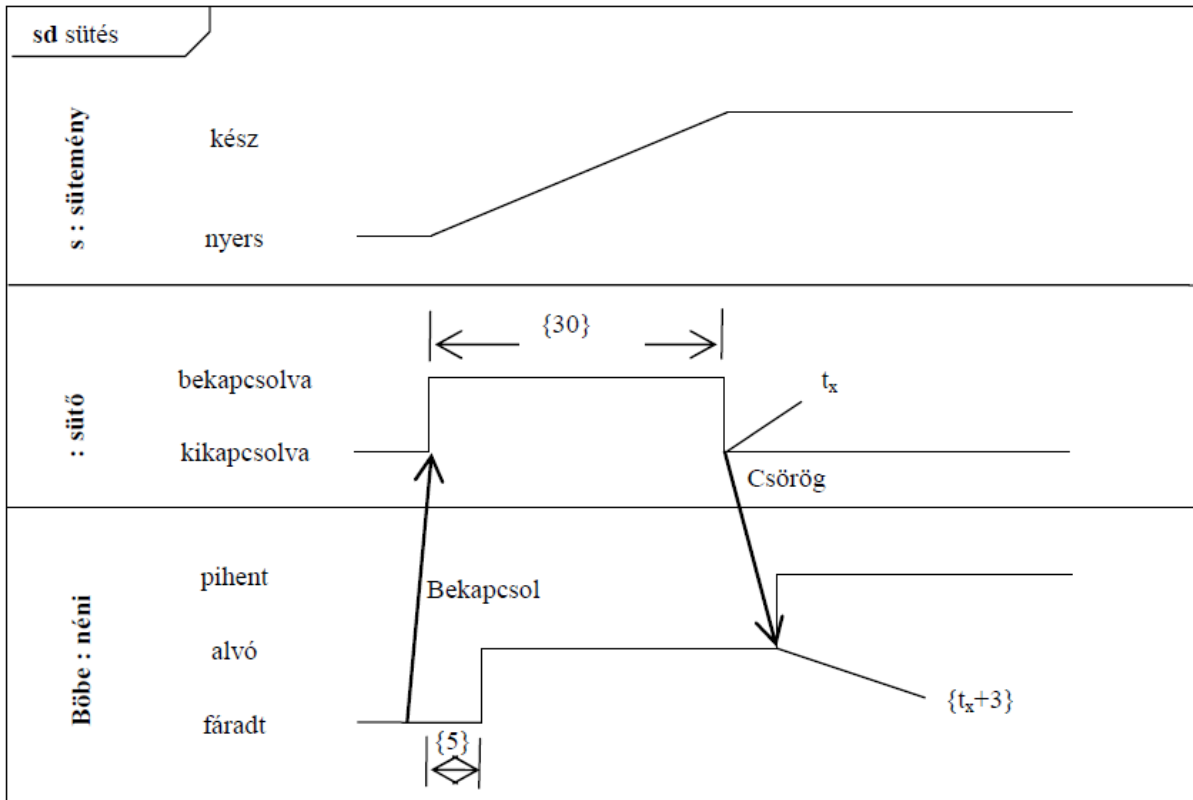
Timing diagram – Megoldások

Timing diagram – Megoldások

2009.01.13 – 9. Feladat

Az alábbi történet alapján rajzoljon UML 2 időzítési diagramot (timing diagram)! (7 pont)

Böbe néni fáradt, de másnapra süteményt kell sütnie. A nyers süteményt beteszi a sütőbe, amit aztán bekapcsol. Öt perc múlva Böbe néni elalszik. A sütő fél óra elteltével kikapcsol és csörög, amire Böbe néni 3 perc múlva kipihenten felébred, és kiveszi a kész süteményt. Böbe néni diszkrét állapotai: fáradt, alvó, pihent. A sütő diszkrét állapotai: ki, be. A sütemény folytonosan változik a nyers és a kész között.

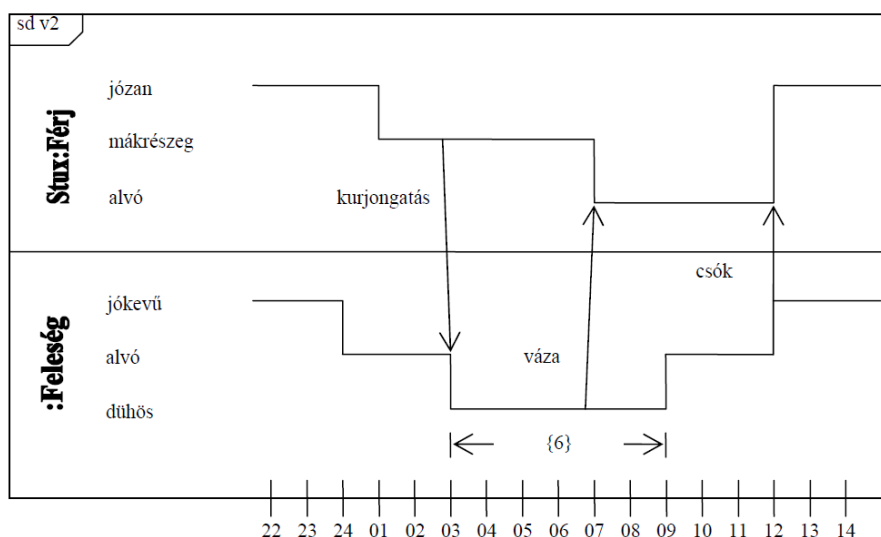


Timing diagram – Megoldások

2010.01.12 (B) – 9. Feladat

Az alábbi történet alapján rajzoljon UML 2.0 időzítési diagramot (timing diagram)! (7 pont)

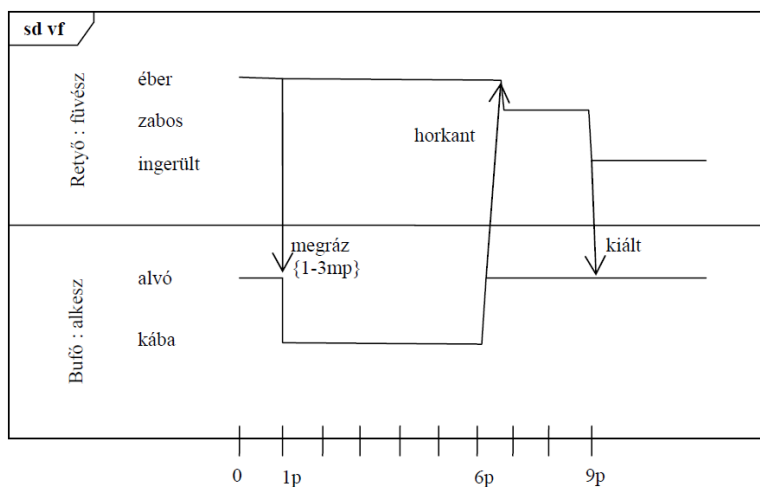
Stux este 10-kor józanul ment el otthonról, egyedül hagyva jókedvű feleségét, aki 12-kor elaludt. Stux hajnali 1-re lett mákrészeg, és hajnali 3-ra ért haza. Ekkor a kurjongatásra felesége dühösen ébred, és hat órán át dühös is marad. 7-kor felesége egy vázát vág hozzá, amitől Stux elalszik. Miután az asszony kidühöngte magát, elalszik. Mikor délben jókedvűen felébred, megcsókolja férjét, aki józanul ébred.



2011.06.14 – 7. Feladat

Az alábbi történet alapján rajzoljon UML 2.0 időzítési diagramot (timing diagram)! (8 pont)

Retyő, a fűvész és Bufó, az alkesz barátok. A történet kezdetekor Retyő éber, Bufó alvó állapotban van. 1 perc után Retyő (1-3 másodpercig) megrázza Bufót, mire az kába állapotba kerül. Bufó 5 perc elteltével visszaalszik, és horkant, amire Retyő zabos lesz. 3 perc múlva Retyő ingerültté válik és Bufóra rákiált, aki erre nem reagál.

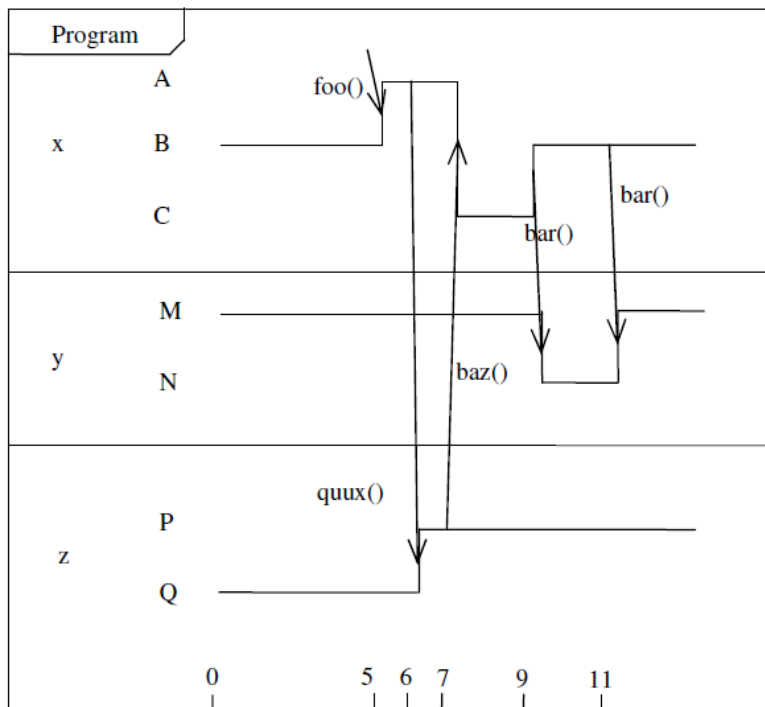


Timing diagram – Megoldások

2013.05.28 – 9. Feladat

Rajzoljon UML 2 időzítési diagramot az alábbi Java kódrészlet alapján az x, y és z objektumokra!

<pre>enum XS { A, B, C } enum YS { M, N } enum ZS { P, Q } class Base { static void sleep(int seconds) { try { Thread.sleep(seconds*1000); } catch (InterruptedException e) { } } } class X extends Base { XS xs = XS.B; void foo() { xs=XS.A; sleep(1); Program.z.quux(); } void baz() { xs = XS.C; sleep(2); Program.y.bar(); xs = XS.B; sleep(2); Program.y.bar(); } } class Y extends Base { YS ys = YS.M; void bar() { ys = ys == YS.M ? YS.N : YS.M; } } class Z extends Base { ZS zs = ZS.Q; void quux() { zs = zs == ZS.P ? ZS.Q : ZS.P; sleep(1); Program.x.baz(); } } public class Program extends Base { static X x = new X(); static Y y = new Y(); static Z z = new Z(); public static void main(String[] args) { sleep(5); x.foo(); } }</pre>	<pre>class Y extends Base { YS ys = YS.M; void bar() { ys = ys == YS.M ? YS.N : YS.M; } } class Z extends Base { ZS zs = ZS.Q; void quux() { zs = zs == ZS.P ? ZS.Q : ZS.P; sleep(1); Program.x.baz(); } } public class Program extends Base { static X x = new X(); static Y y = new Y(); static Z z = new Z(); public static void main(String[] args) { sleep(5); x.foo(); } }</pre>
---	--

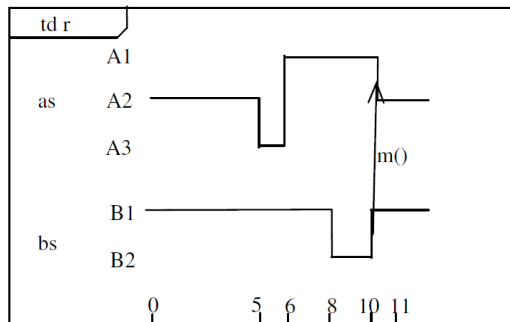


Timing diagram – Megoldások

2014.01.14 – 6. Feladat

Rajzoljon UML2 timing diagramot az **r** objektumhoz!

<pre>enum AS { A1, A2, A3 } enum BS { B1, B2 } class R { private AS as = AS.A2; private BS bs = BS.B1; public void m() { if (bs == BS.B1) { switch (as) { case A1: as = AS.A2; break; case A2: as = AS.A3; break; case A3: as = AS.A1; break; } } } public void t() { if (bs == BS.B1) {bs = BS.B2; } else {bs = BS.B1; m();} } }</pre>	<pre>public class Program { public static void main(String[] args) { R r = new R(); s(5); r.m(); s(1); r.m(); s(2); r.t(); s(2); r.t(); s(1); } private static void s(int c) { try { Thread.sleep(c*1000); } catch (InterruptedException e) {} } }</pre>
--	---



asd